1988

# Polar Spectrum Coding

Daniel Harris Chapman
*University of Central Florida*

UNIVERSITY OF CENTRAL FLORIDA

OFFICE OF GRADUATE STUDIES

THESIS APPROVAL

DATE: April 5, 1988

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION

BY Daniel H. Chapman

ENTITLED "Polar Spectrum Coding"

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE

DEGREE OF Master of Science in Engineering

FROM THE COLLEGE OF Engineering

Yacoub A. Alsaka
Supervisor of Thesis

RECOMMENDATION CONCURRED IN:

Brian Petrasko

Bruce E. Mathews
Coordinator of Degree Program

COMMITTEE ON FINAL EXAMINATION

Louis M. Trefonas
Dean of Graduate Studies

# POLAR SPECTRUM CODING

BY

DANIEL HARRIS CHAPMAN
B.S., Christian Brothers College, 1981

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in the Graduate Studies Program
of the College of Engineering
University of Central Florida
Orlando, Florida

Spring Term
1988

# ABSTRACT

Polar Spectrum Coding is a novel speech coding algorithm for narrowband voice communications. A polar Fourier transform of the signal is computed, and the magnitude and phase of the speech spectrum is encoded for transmission. The correlation between frames of speech signals is exploited to minimize the transmission rate required for intelligible speech. At the receiver, the encoded words are decoded and the spectrum reconstructed. An inverse Fourier transform is performed, and the result is the reconstructed speech waveform.

Polar Spectrum Coding theory is explained. The sensitivities of various parameters on performance are explored, and performance in the presence of channel noise is measured. Directions for future research in the realm of Polar Spectrum Coding is suggested.

To Cathy,

for her encouragement and her love,

and Amanda,

who unknowingly sacrificed precious time together

so that Daddy could play on the computer

## ACKNOWLEDGEMENTS

I gratefully acknowledge the help of Dr. David Chester and Dr. Dennis Tebbe of Harris Corporation in gathering ideas for the research in this thesis. Special thanks goes to Mark Webster of Harris Corporation for his enthusiastic help, answers to tough questions, and his ability to keep me focused on the basic problem at hand. I am indebted to Dr. Yacoub Alsaka, both for his support and for joining me on this journey of exploration.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

## LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

| | |
|---|---|
| PSC | Polar Spectrum Coding |

GLOSSARY

Center frequency. The center frequency is the frequency bin at which the center of the magnitude corrector is positioned. If the magnitude corrector has an even number of frequency bins, then the center frequency is defined as the closest frequency bin below the center of the magnitude corrector.

Corrector. A corrector is a group of data bits that represents the change that will be made to the reference spectrum of the next frame to cause it to be different from the reference spectrum of the present frame. In the case of a single corrector per frame, the corrector is made to be as close as possible to the error spectrum. In the case of multiple correctors in each frame, the summation of the correctors in a frame is made to be as close as possible to the error spectrum.

Error magnitude spectrum. The difference between a signal magnitude spectrum and a reference magnitude spectrum is the error magnitude spectrum.

Frequency bin. A frequency bin is a discrete frequency component of a spectrum.

Gain. The gain of a magnitude corrector is the magnitude of the magnitude corrector at its maximum magnitude.

Highest phase bin. The highest phase bin is the value of the phase of the highest frequency bin of the spectrum.

Lowest phase bin. The lowest phase bin is the value of the phase of the lowest frequency bin of the spectrum.

Magnitude corrector. A magnitude corrector is the magnitude portion of a corrector.

Magnitude corrector sign. Each magnitude corrector must either add to or subtract from the reference magnitude spectrum of the present frame to shape the reference magnitude spectrum of the next frame. The magnitude corrector sign determines which operation is performed.

Phase bin. Each frequency bin of a spectrum has an associated phase value. Each frequency bin, as it relates to phase, is called a phase bin. An FFT of a signal provides a discrete number of frequency bins. The phase value of a particular frequency bin is the value of the phase bin.

Reference magnitude spectrum. The synthesized speech spectrum of each frame is the reference magnitude spectrum. There is one reference magnitude spectrum for each signal frame.

Shape. The envelope of the corrector is its shape. For example, a triangular corrector has its highest (or lowest) magnitude at its center frequency.

Signal frame. A group of successive quantized samples of a speech signal is a signal frame. In most speech processing literature, the number of samples in a frame is designated by N. PSC is performed on one signal frame of a speech signal at a time.

Start frequency. The start frequency is the frequency bin at which the lowest frequency bin of the magnitude corrector is positioned.

Width. The width of a magnitude corrector is the number of frequency bins that the magnitude corrector spans, including the frequency bins at both ends of the magnitude corrector. The width is a measure of the bandwidth of the corrector.

# CHAPTER 1

# INTRODUCTION

Speech coding is the process of encoding a speech signal with a minimum number of bits to be transmitted over a channel (and hence the transmission rate), and decoding the received bits into a reconstructed speech signal. There are two primary goals in narrowband speech coding, and a particular coding system might serve to realize either or both goals. The first goal is to minimize the transmission rate required for a given minimum speech intelligibility or speaker recognizability threshold. The second goal is to maximize the intelligibility or speaker recognizability for a given transmission rate. Although many speech coding algorithms attempt to reach both primary goals, often the emphasis is on one of the goals.

The constraints of low tranmission rate and high voice quality are conflicting requirements. The quality of the signal can be as high as desired, provided that the bandwidth required is of no concern. Conversely, the transmission rate can be lowered to any arbitrarily low value at the expense of speech quality.

Polar Spectrum Coding (PSC) is aimed at minimizing the transmission rate required for intelligible speech. To this end, attempts are made to expose the most important information contained in a speech signal. The critical information is then efficiently encoded for narrowband communications. Irrelevancy and redundancy in a signal provide opportunities for bandwidth compression (Jayant and Noll 1984, 16). Irrelevancy in a signal is the information contained in the signal that is imperceptible to the intended receiver (usually a person in the case of speech). Irrelevancy can be eliminated from the signal without any perceptible degradation to the signal. Redundancy in a signal by definition contains no new information, but might be important to the perception of the signal. Therefore, redundancy in a signal should be preserved; but it is not necessary to transmit the redundancy itself. The goal of PSC, therefore, is to throw out the irrelevancy of a signal and efficiently encode the redundancy such that the required transmission rate is minimized while the critical components of the signal are preserved.

PSC is shown to be a speech coding algorithm that has the potential of delivering narrowband speech communication efficiently. The quality of the speech obtained depends on the allowable tranmission bit rate, but a summary of current PSC performance is as follows. A transmission rate of 1000 bits per second gives intelligible but severely degraded speech. Individual words are often difficult to pick out without the benefit of surrounding words for clarity. At 4800 bits per second, individual words are much clearer; and PSC provides synthetic quality speech. Communication quality speech is obtained at 9600 bits per second.

Chapter 2 compares several common speech processing algorithms in use today. These range from simple to complex, require from low to high bit rates, and provide from synthetic to broadcast quality speech. Various performance measures for the algorithms are discussed. Chapter 3 details the theory behind PSC. Design tradeoffs are discussed, as well as the search for the most critical information in speech waveforms. Chapter 4 presents results obtained with PSC. The effects of various parmeters on the quality of speech are shown. Measurements of performance in the presence of noise are presented.

PSC is an interesting field of study, not only because it is a novel technique of encoding speech, but because it opens many opportunities for further research. In fact, the research documented here uncovers many more questions than it answers. Chapters 3 and 4 should provide the reader with many avenues of pursuit in search of improved performance.

# CHAPTER 2

# COMPARISON OF SPEECH CODING ALGORITHMS

## Speech Coding Algorithms

Many speech coding algorithms are in common usage today, and many more algorithms have been proposed. Most algorithms can be classified as either time domain processing or frequency domain processing. Time domain processing implies that the speech signal is processed without the use of Fourier transforms; frequency domain methods require Fourier transforms somewhere in the algorithm. Alternatively, algorithms can be classified as either waveform coding or analysis/synthesis coding. Waveform coding attempts to preserve the shape of the waveform. In the frequency domain, the "waveform" is the spectrum of the signal. Analysis and synthesis coding, on the other hand, does not have waveform preservation as its goal; its purpose is to generate a speech signal which is perceptually similar to the original. Figure 1 outlines an organization of several of the most common speech processing algorithms. Time domain waveform coders include Pulse Code Modulation, Differential Pulse Code Modulation, Delta Modulation, and Sub-Band Coding. Sub-Band Coding has the distinction of also falling under the frequency domain waveform category. Transform Coding

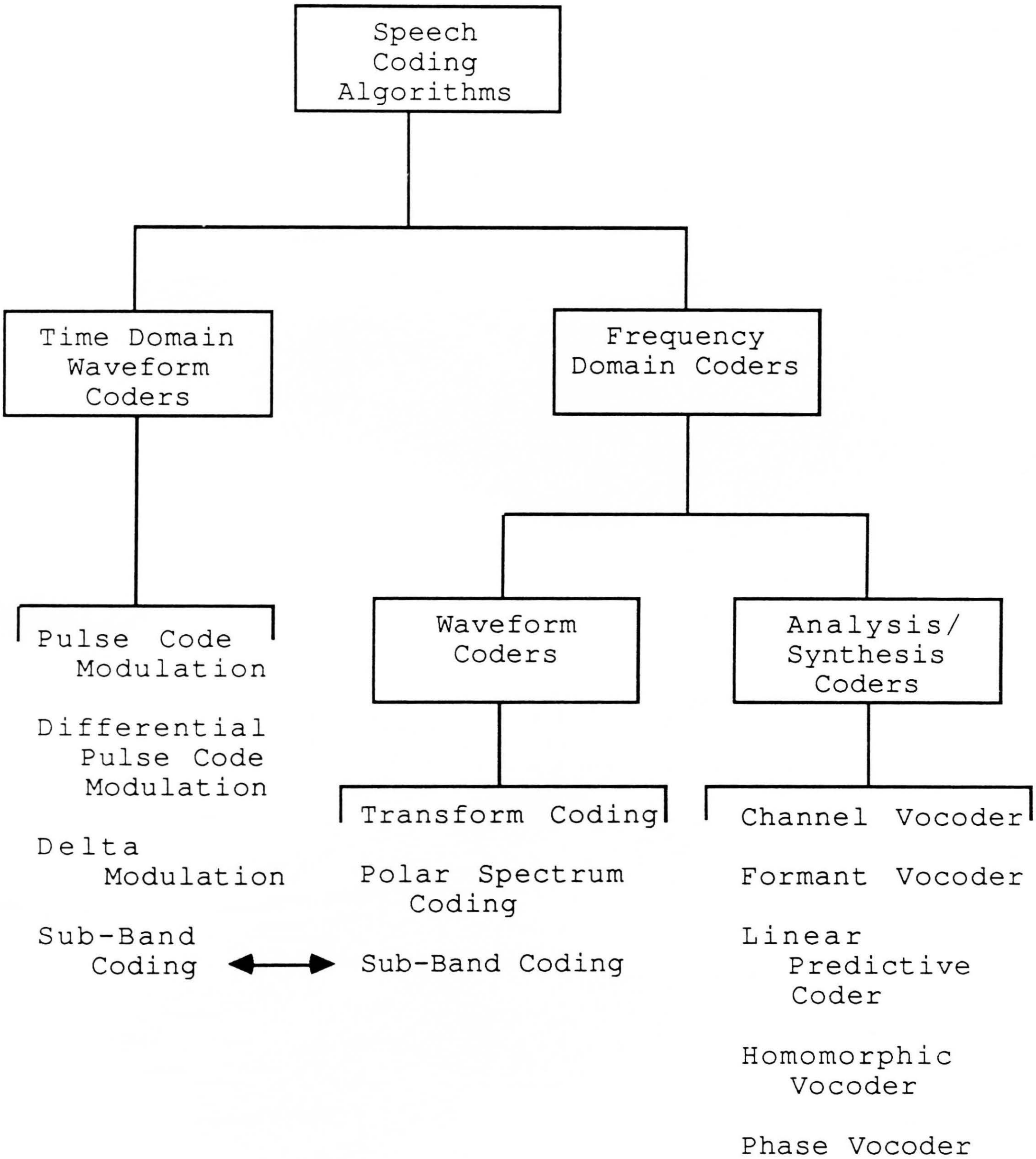



Figure 1. Classifications of Speech Coding Algorithms.

and Polar Spectrum Coding are frequency domain waveform coding techniques. The various vocoders are generally classified as frequency domain analysis/synthesis algoritms. Vocoders include the Channel Vocoder, the Formant Vocoder, Linear Predictive Coding, the Homomorphic Vocoder, and the Phase Vocoder. Each of these coding techniques are described in later sections of this chapter, except for Polar Spectrum Coding. Chapter 3 is devoted to a detailed explanation of Polar Spectrum Coding.

All digital speech processing systems have certain components in common. First, the incoming analog signal is lowpass or bandpass filtered to remove unwanted frequency components. The most common filter is a lowpass filter which is used to eliminate aliasing during the sampling process. Second, the analog signal is sampled and quantized into discrete levels at discrete time periods. Third, the reconstructed signal at the decoder output is filtered to eliminate the extra frequency components which contain frequency-shifted replicas of the signal spectrum due to the sampling process. These components of the system are assumed in the following discussions.

# Pulse Code Modulation

Pulse Code Modulation (PCM) is the simplest method of coding speech signals. A number of bits (typically eight or twelve) are alloted to each sample of speech to represent its value. The data bits are transmitted by the encoder and received by the decoder. A digital to analog conversion translates the bits back into a reconstructed waveform. Figure 2 shows a block diagram of a PCM system.

Several issues are of importance in a PCM coding algorithm. Among them are the number of quantization levels, the difference between the quantization levels, and word synchronization. N quantization levels requires $\log_2 N$ bits to specify each sample value. Quantization levels can be uniform in size or varying. Non-uniform levels provide more dynamic range for the number of bits allocated. Quantization error is the difference between the quantized value and the original value. In order to minimize the quantization error, the quantization step size(s) should be small; but step sizes must be large enough to track wide dynamic ranges of the signal. Therefore, there is a tradeoff that must be made in the design of quantization step sizes. The decoder must determine the framing of the sample words so that it can correctly decode each word. One method used to solve the problem involves the use of a special synchronization word which is sent at the beginning of a transmission. The

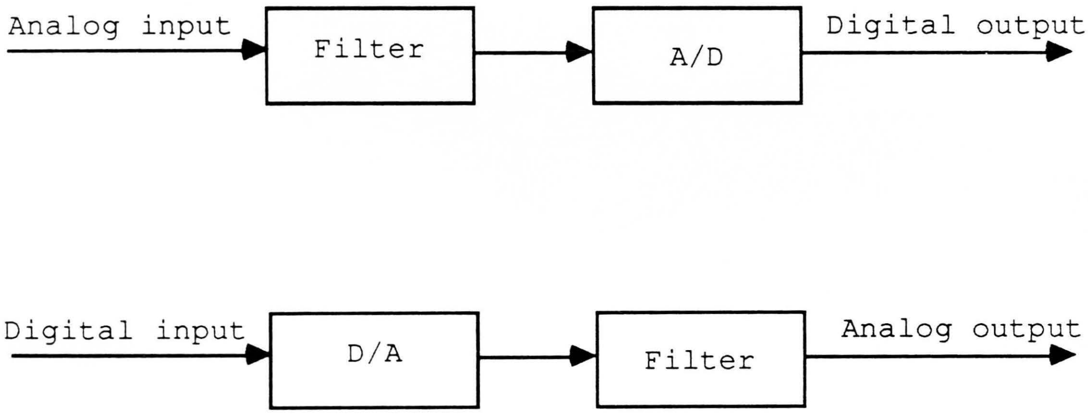



Figure 2. PCM System. Encoder and Decoder.

receiver searches for the synchronization word to determine the timing of the sample words.

## Differential Pulse Code Modulation

Differential Pulse Code Modulation (DPCM) uses the techniques of PCM but requires a narrower channel bandwidth. It has long been understood that speech signals contain a large measure of redundancy. In order to more efficiently encode the speech signal, the difference between successive samples can be encoded and transmitted. This reduces the required bandwidth (or the quantization step size) because there is a high correlation between successive speech samples, especially when the speech signal is oversampled (sampled at greater than the Nyquist rate).

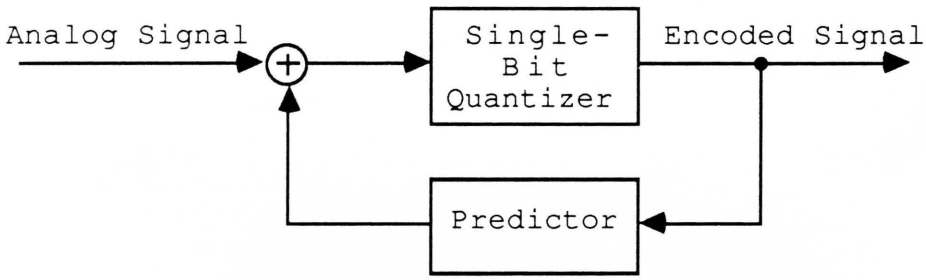
Adaptive quantization levels allow the signal to be encoded with fewer bits for the same dynamic range. The actual value of a quantization level in adaptive quantization depends on previous values. Adaptive quantization, therefore, introduces the concept of memory into speech coding algorithms.

Two primary types of errors that DPCM must try to minimize are granularity noise and slope overload. Granularity error is the inability of the system to specify a difference as small as the change in the signal. Granularity error is most prevalent during slowly varying signal amplitude levels. The distortion caused by slope overload, by contrast, is most prevalent during times of fast signal amplitude changes. If the coder is not able to indicate the change of the signal as fast as the slope of the signal requires, slope overload occurs.

Differential systems of all types are more sensitive to transmission errors than non-differential systems, unless they are carefully designed. Since a sample value at the decoder output depends on previous values, an error in channel transmission will propagate indefinitely. One way to reduce the sensitivity of differential coding to bit errors is to include error correction coding on the size of the quantization step (Papamichalis 1987, 37).

## Delta Modulation

Delta Modulation (DM), shown in Figure 3, is a DPCM system in which one bit is used to indicate the difference value of each sample. The value of the bit indicates whether the value of the sample is higher or lower than the previous sample. Since there is only one bit transmitted per speech sample, the signal must be oversampled at a rate much higher than the Nyquist rate in order to obtain a high correlation between samples. Linear Delta Modulation (LDM) uses a fixed level for the quantize step value; Adaptive Delta Modulation (ADM)



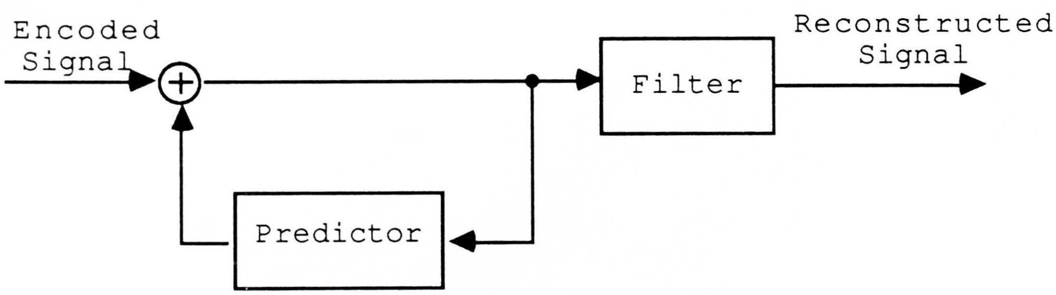




Figure 3. DM System. Encoder and Decoder.

varies the step size to track the speech signal waveform. DM, like DPCM, must contend with slope overload and granularity noise. Slope overload minimization implies large quantization step sizes, while granularity noise minimization implies small quantization step sizes. Careful tradeoffs must be made concerning the quantization step sizes in order to achieve top performance of a DM system.

A special case of DM is Continuously Variable Slope Delta modulation (CVSD). CVSD improves the performance of a DM system in the presence of transmission errors. Whenever the present bit value matches the value of the previous two bits, the step size is increased. Otherwise, the step size is reduced. The price for improved performance in the presence of channel noise is that CVSD has the drawback of degraded speech quality (Papamichalis 1987, 47).

## Sub-Band Coding

It is known that the perception of the quality of a speech signal does not depend on the noise in the signal alone. One important dependence on perceptual quality concerns the frequency range of interest. In other words, noise in a speech signal degrades the quality of the speech in varying amounts, depending on the frequency range that the noise affects. This implies that signal information is more critical in some frequency ranges than in others; and this, in turn, implies that more bits should be used to encode certain frequency ranges of signals than others. Sub-Band Coding (SBC) is a system that exploits these properties of a speech signal. A block diagram of a SBC system is shown in Figure 4.

In SBC, the input analog speech signal is sent through a parallel bank of bandpass filters to split the signal into several frequency bands. Each band is translated to a baseband frequency signal and sampled at the Nyquist rate of the baseband bandwidth or higher. Each frequency band is then encoded using adaptive PCM, ADM, CVSD, or another suitable algorithm.

The efficency in SBC is gained in the encoding process of each frequency band. More bits are used for critical bands, and less bits for less critical bands. In addition, if transmission errors are a problem, error correction coding can be employed only on the bands that are most critical. This reduces the overhead required for error correction.

## Transform Coding

Transform Coding (TC) represents a class of coding algorithms in which a block of input samples are transformed through a linear mapping into a set of coefficients. The goal of the transformation is to

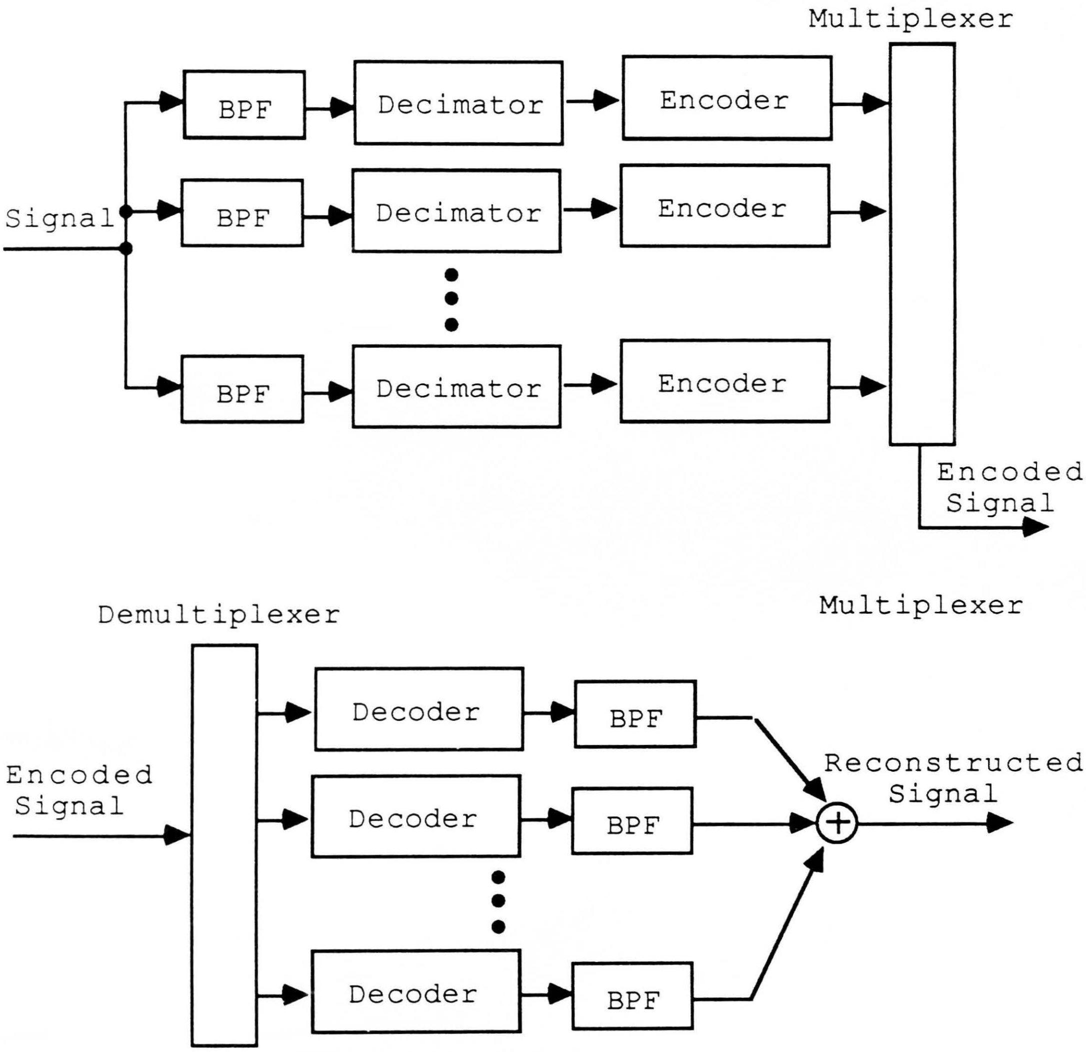



Figure 4. Sub-Band Coding System. Encoder and Decoder.

generate a set of coefficients that are as uncorrelated as possible (Jayant and Noll 1984, 514). This decorrelation removes the redundancy from the signal and, therefore, minimizes the required bit rate. A generalized block diagram of a TC system is shown in Figure 5. In the encoder, the samples are processed through a transform matrix, and the resulting coefficients are transmitted. Adaptive bit allocation allows the bit structure to change from frame to frame, according to the requirements of the changing speech waveform signal properties; this type of system is called Adaptive Transform Coding (ATC). In the decoder, an inverse transform takes place to reconstruct the waveform samples.

There are many transforms that can be implemented in a TC system. Among them are the Discrete Cosine Transform (DCT), Fourier Transform, Walsh-Hadamard, and Karhunen-Loeve Transform. A transform type is judged by how well it removes redundancy from the speech signal and by how easily the transform is implemented. The DCT is one of the best of the practical transforms both in terms of performance and in terms of simplicity (Papamichalis 1987, 79).

## Analysis/Synthesis Coders

All of the coding systems described above belong to the general class of waveform coders because they attempt

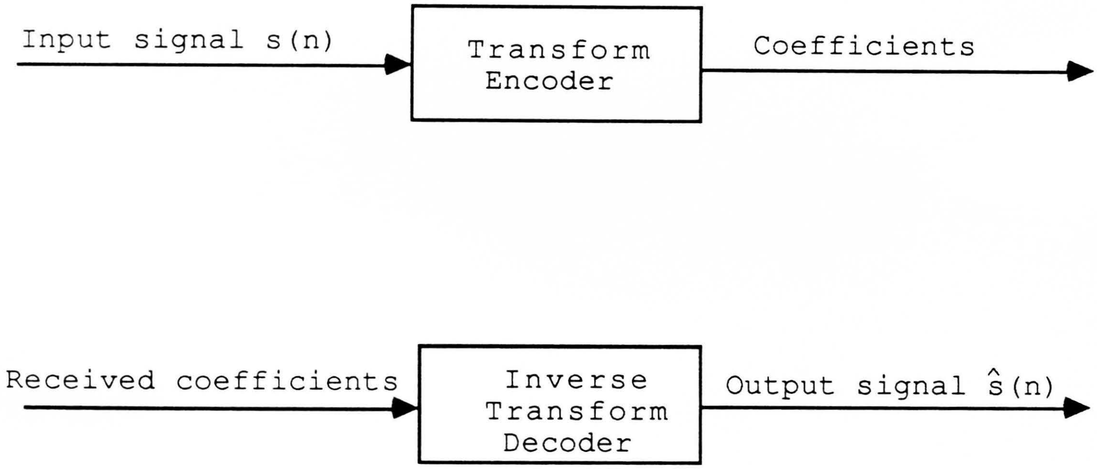



Figure 5. Simplified Model of a Transform Coder.

to reconstruct the actual waveform, in either the time or frequency domain, of the original signal. Another class of coders, known as analysis/synthesis coders, source coders, or vocoders, attempts to create a signal that is a perceptual equivalent to the original speech signal. Vocoders try to imitate the properties of the vocal tract of the speaker. Good quality speech can be attained with vocoders at bit rates between 2400 and 9600 bps. Since this class of coders is not closely related to PSC (with the exception of the phase vocoder), only a brief descrption of several types of vocoders will be given.

Channel Vocoder. The channel vocoder is the oldest vocoder design. Figure 6 shows a channel vocoder. The speech signal is bandpass filtered into parallel frequency bands, and the envelope of each band is determined. The envelope of each band is encoded for transmission. Also transmitted are a voice/nonvoice bit and, in the case of voiced segments of speech, the pitch. The decoder uses the voice/nonvoice bit to determine the excitation of the received frequency bands. If the speech is voiced, then a pulse source with the pitch frequency is applied to the bands; nonvoiced segments are excited with a noise source. After filtering to remove high-frequency spectral images, the bands are summed together to form the output signal.

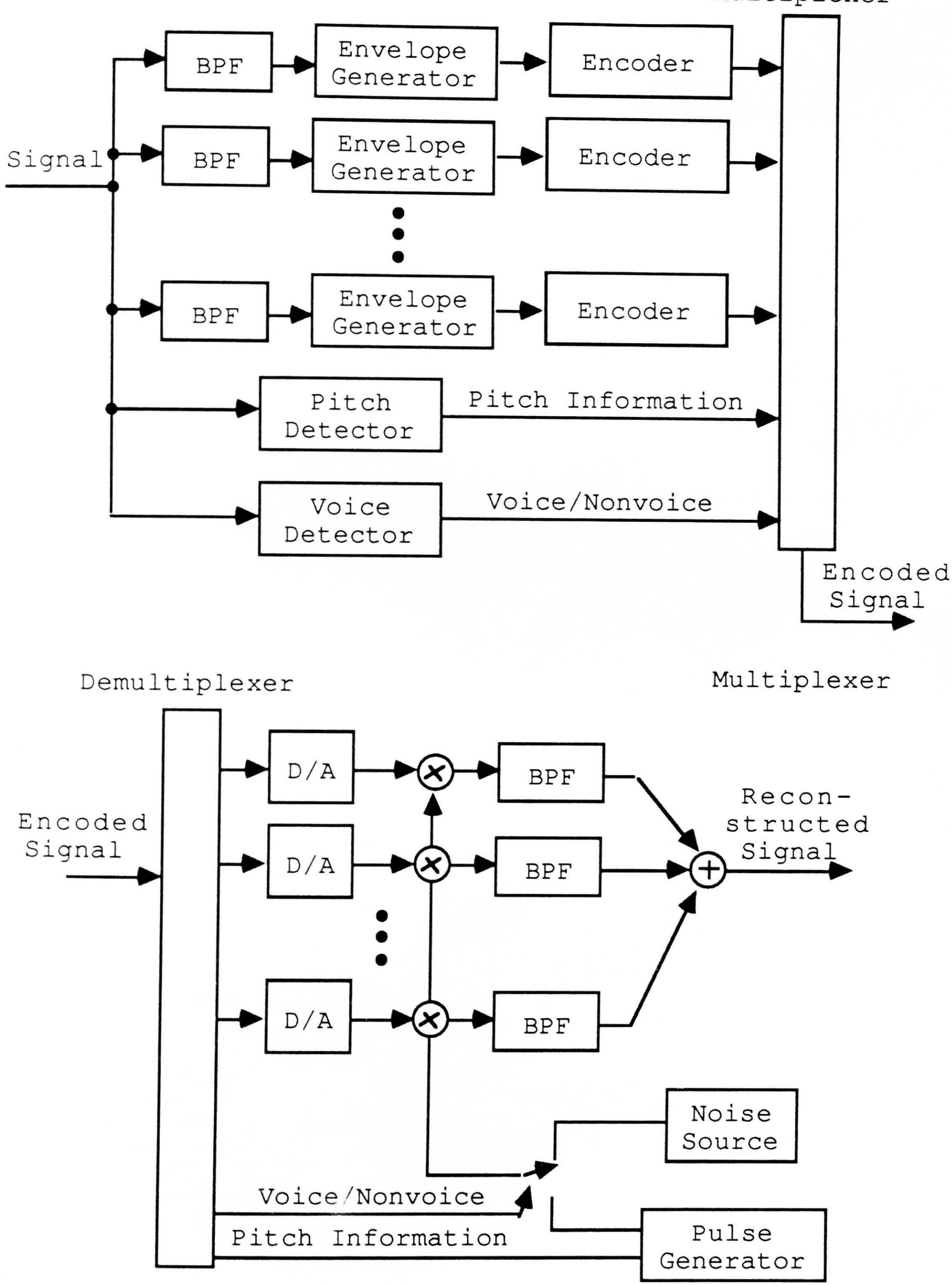



Figure 6. Channel Vocoder System. Encoder and Decoder.

Formant Vocoder. The formant vocoder uses the concepts found in the channel vocoder, with the primary addition that formant frequencies are also transmitted. Formants are the most important elements affecting the perceptual quality of speech signals (Papamichalis 1987, 102). The center frequencies of the first few formants, usually three, are determined by the encoder. The bandwidths of the formants are not transmitted to the decoder; instead, the decoder gives each formant a fixed bandwidth that is based on the center frequency of the formant. The bandwidths given to the formants are derived by experimentation and are considered to be typical for the types of speech that a particular formant vocoder is likely to process.

Linear Predictive Coding. Linear Predictive Coding (LPC) is the most popular vocoder, and one of the most popular of all speech coders. LPC has been experimented with by so many people that there are many versions, adaptations, and extensions available to a designer. Its popularity is due to its performance, especially at low bit rates.

LPC contains the same basic components as a channel vocoder. The unique aspect of LPC is that it models the vocal tract as an all-pole system, using weighted values of past samples to predict each new prediction coefficient. The poles are selected to minimize the mean squared error between the all-pole filter frequency response and the envelope of the speech spectrum. Nasal voiced sounds and fricatives normally require both poles and zeros for adequate representation, but an all-pole system is accurate enough if the order of the filter is high enough (Rabiner and Schafer 1978).

Homomorphic Vocoder. The homomorphic vocoder performs a homomorphic transformation on the speech signal (Oppenheim and Schafer 1975). The Fourier transform of the speech signal is computed. The logarithm of the magnitude of the spectrum is taken, and then the inverse Fourier transform is obtained. The result is called the cepstrum of the signal. The cepstrum separates the slow-changing envelope of the spectrum from the fast-changing slopes due to the pitch of the signal. The two components are separately encoded for transmission and decoding. In the decoder, an inverse transformation is performed on the encoded cepstrum information.

Phase Vocoder. The phase vocoder is perhaps a closer relative to PSC than any of the previously mentioned vocoders. A phase vocoder encodes both magnitude and phase spectrum information for transmission. The part of the phase vocoder that encodes magnitude information is similar to the channel vocoder. Phase vocoders take advantage of the fact that the phase of a particular frequency component changes smoothly over time (Papamichalis 1987, 99). The derivative of the phase is computed and transmitted.

## Performance Measures of Speech Coding Algorithms

When a comparison is made between two or more speech coding algorithms, certain qualities of the algorithms are easily compared. The easily compared aspects include transmission bit rate, computational complexity, and hardware cost. Perhaps the most important quality of a speech coding algorithm, however, is its performance. Unfortunately, this aspect of a speech coding system is probably the most difficult to measure. In fact, there is no "best" way to measure the performance of a speech coding algorithm; instead, there are several ways, each of them having merit. These performance measurement techniques fall into two categories: objective and subjective. Each is discussed at length.

### Objective Performance Measures

Objective performance measures seem to hold the promise of a true, unbiased judgment of all coding systems. Indeed, this may one day be the case. New objective performance tests are being investigated, and one of these techniques might be just the test researchers are looking for. Until that time comes, however, researchers must rely on a set of objective performance measures that do not accurately measure the quality of the speech as perceived by the ear and brain. In spite of the drawbacks of objective performace measures, they are widely used because they are easy to compute, do not rely on listener biases, and in many cases provide good indications of the quality of the speech.

Signal to Noise Ratio. The signal to noise ratio (SNR) is the most popular objective measure of the performance of a speech coding system. The SNR of a waveform is the ratio of the signal error variance to the variance of the original signal. It is usually expressed in decibels.

The error signal, e(n), is defined as the difference between the original signal and the reconstructed signal:

$$e(n) = s_{orig}(n) - s_{reconstructed}(n).$$

The error signal must be computed under three conditions. The first is that both the original and reconstructed signals have zero mean. Otherwise, the error signal will have a DC offset that is imperceptible in speech but will adversely affect later calculations. The second condition is that both signal waveforms must be time aligned for maximum correlation. In other words, if the reconstructed signal is time shifted, then the shift must be taken out before the calculation is made. Time shifts, like DC offsets, do not affect the quality of the speech and therefore should not affect the performance measures. The third condition under which the error signal must be computed involves the relative energies of the two signal waveforms. A difference in energy is perceived as a change in the loudness of the speech and, within reason, does not affect the quality of the speech. Therefore, the energy of the reconstructed speech signal should be normalized to the energy of the original speech before the error waveform is calculated.

Once the error signal is computed, under the conditions of zero mean, zero time shift, and normalized energies, the energy of the original signal and the error signal are calculated:

$$E_s^2 = (1/N) \sum_{n=1}^{N} s^2(n) \text{ and}$$

$$E_e^2 = (1/N) \sum_{n=1}^{N} e^2(n) \ .$$

The SNR is expressed as

$$\mathrm{SNR(dB)} = 10 \log_{10}(E_s^2/E_e^2) \ .$$

<u>Segmental Signal to Noise Ratio.</u> The SNR value of a signal is dominated by the segments of speech that contain high levels of energy. The segmental SNR (SNRSEG) is used to give a more perceptually accurate measure of a typical speech signal (Kitawki et al. 1982). This is particularly true for signals in which some segments of the speech

contain much less energy than others. In calculating the SNRSEG of a waveform, the signal is divided into segments of equal length, typically 16 ms (Papamichalis 1987, 180). The SNR is calculated for each segment independently. The SNRSEG is the mean value of the SNR of all of the segments:

$$\mathrm{SNRSEG(dB)} = (1/M) \sum_{m=1}^{M} \mathrm{SNR(m)(dB)} \quad ,$$

where m is the index of each frame.

Frequency Weighted Signal to Noise Ratio. Perceptually, distortion in certain frequencies of a speech signal is more objectionable than in other frequencies. Although the perception level for various frequencies will be different for each person, there is a general trend that makes frequency weighting of the SNR calculation a more accurate measurement than the SNR without the frequency weighting. The articulation index, given by Jayant and Noll (1984), divides up the spectrum into frequency bands of equal perceptual importance. The SNR is weighted according to these frequency bands.

Spectral Distance Measures. Spectral distance measures calculate the distance between some function of the original spectrum and the spectrum of the reconstructed waveform. These measures are used on vocoders, since the output of a vocoder system is often a signal that is perceptually close to the original signal but has a very different waveform shape. Several examples of spectral distance measures are log spectral distance, cepstral distance, log likelihood ratio, LPC spectral distance, cosh measure, and Euclidean distance.

## Subjective Performance Measures

Subjective performance measures involve grading of a speech coder output by listeners as they hear the speech. The precision and repeatability of subjective performance measures do not match those of objective measures. Personal biases, differences in grading levels, and fatigue must be dealt with when subjective measures are used. Subjective performance measures require much more time and effort than objective measures. The advantage that subjective measures have over objective measures is that they more accurately measure the perceived quality of speech signals. Since the perceived quality of speech is the most important attribute of speech in communication, subjective tests are performed, in spite of the time and effort involved, when a detailed evaluation of a speech coding system is desired. Subjective measures primarily involve intelligibility tests or quality preference tests.

Intelligibility Tests. Intelligibility tests involve pairs of words which have one phoneme difference between them, such as the pair {fin, thin}. One word is spoken, and the listener must determine which word of the pair was spoken. Since the listener is faced with only a binary decision, the amount of subjectivity in an intelligibility test is minimized. As a consequence, intelligibility test scores have variances that are smaller than some other types of subjective tests. There are several versions of intelligibility tests, but the most popular is the Diagnostic Rhymne Test (DRT). A DRT score P is given by

$$P = 100(R - W)/T ,$$

where R is the number of right answers, W is the number of wrong answers, and T is the total number of word pairs tested.

Quality Preference Tests. Quality preference tests contain more subjectivity than intelligibility tests. The measurement of quality preference tests is not intelligibility, but quality, of the speech signal. It is possible that, if samples of two speech coders are presented to two listeners, one listener will consistently prefer system A while the other listener will consistently prefer system B. This could be due to the fact that hissing noises annoy the first listener and buzzing sounds are objectionable to the second listener. The Diagnostic

Acceptability Measure requires listeners to rate sentences, on a scale of between 0 and 100, on various characteristics of the speech. Another method, using Mean Opinion Scores, requires listeners to grade speech signals on a scale of between 1 and 5.

## Comparison of Performance Measures

Experiments have been performed to determine the quality of objective performance measures compared to subjective performance measures (Kitawki et al. 1982). Results are shown in Table 1. The performance criterion is the variance of objective measures given a specified Mean Opinion Score; lower variance results in a better measure. As the table shows, SNRSEG is a better measure than the standard SNR. Frequency domain measures are better than time domain measures, though not by much.

TABLE 1

COMPARISON OF OBJECTIVE AND SUBJECTIVE PERFORMANCE MEASURES.

| OBJECTIVE MEASURE | STANDARD DEVIATION |
|---|---|
| Signal to Noise Ratio | .665 |
| Segmental SNR | .576 |
| LPC Cepstrum Distance | .458 |
| Cosh Measure | .472 |
| Likelihood Ratio | .477 |

(Data taken from Kitawki et al. 1982.)

## Performance of Speech Coding Algorithms

Various comparative tests have been performed on speech coding algorithms. There is some difficulty in the determination of which coder system is "best," because each algorithm has its own requirements concerning transmission bit rate, complexity, and quality of speech. However, quantitative comparison data have been gathered through experimentation. Figure 7, for example, compares the subjective performance of various general categories of coding systems. Vocoders provide the best speech quality at transmission bit rates lower than about 8 kbps. However, as the bit rate increases, waveform coders provide better performance because vocoders generally do not have a Mean Opinion Score higher than 3.0 (Jayant and Noll 1984).

Figure 8 shows the results of objective performance tests for several waveform coding systems. Vocoders are difficult to measure objectively because they do not attempt to duplicate the original waveform; therefore, they are not included in the figure. As seen in the figure, ATC gives the best SNR over the transmission bit rate range of between 9 kbps and 24 kbps. DPCM provides the worst performance under 16 kbps. The performance of SBC is seen to be less dependent on the transmission bit rate as the other methods.

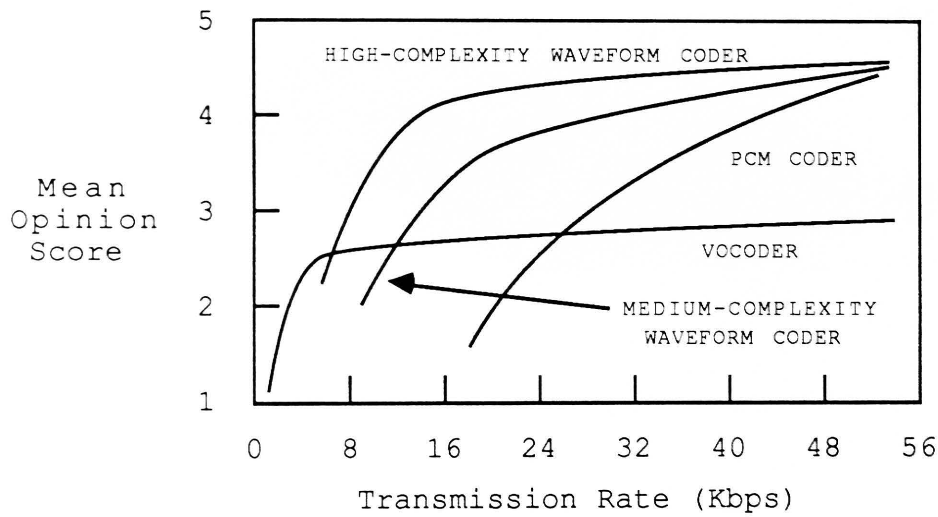



Figure 7. Mean Opinion Scores for Speech Coding Algorithm Categories as a Function of the Transmission Rate. (Data taken from Jayant and Noll 1984.)

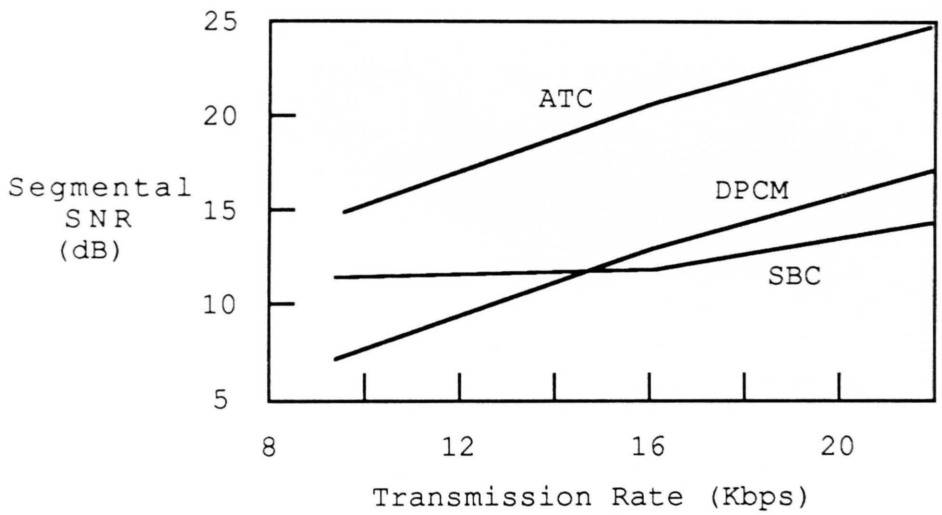



Figure 8. Segmental SNR as a Function of the Transmission Rate. (Data taken from Jayant and Noll 1984.)

# CHAPTER 3

# POLAR SPECTRUM CODING

Digital transmission of speech signals requires that a compromise be made between the two conflicting attributes of good speech reproduction and low transmission bit rate. At one extreme, almost perfect speech reproduction can be obtained at the expense of a data bit rate of 100 Kbps or higher. At the other extreme, vocoders utililizing a channel bandwidth of 50 bps or lower are available; the price paid is intelligible but very synthetic-sounding speech. Many types of systems fall into the vast area between the two extremes. Some systems attempt to preserve the speech waveform, and others attempt to generate a signal that is perceptually equivalent to the original speech. Linear predictive coding, in particular, has gained widespread acceptance as a suitable representation of speech for digital transmission because of its ability to make a good compromise between good speech reproduction and low bit rate. There is a continuing market for communications systems which provide adequate communication but require a minimum of transmission bandwidth. Therefore, a system which has the potential of simultaneously transmitting good quality speech and requiring only a narrow channel bandwidth is worth investigation. The system described in this thesis, which is given the name of Polar Spectrum Coding, is possibly such a system.

## Polar Spectrum Coding Theory

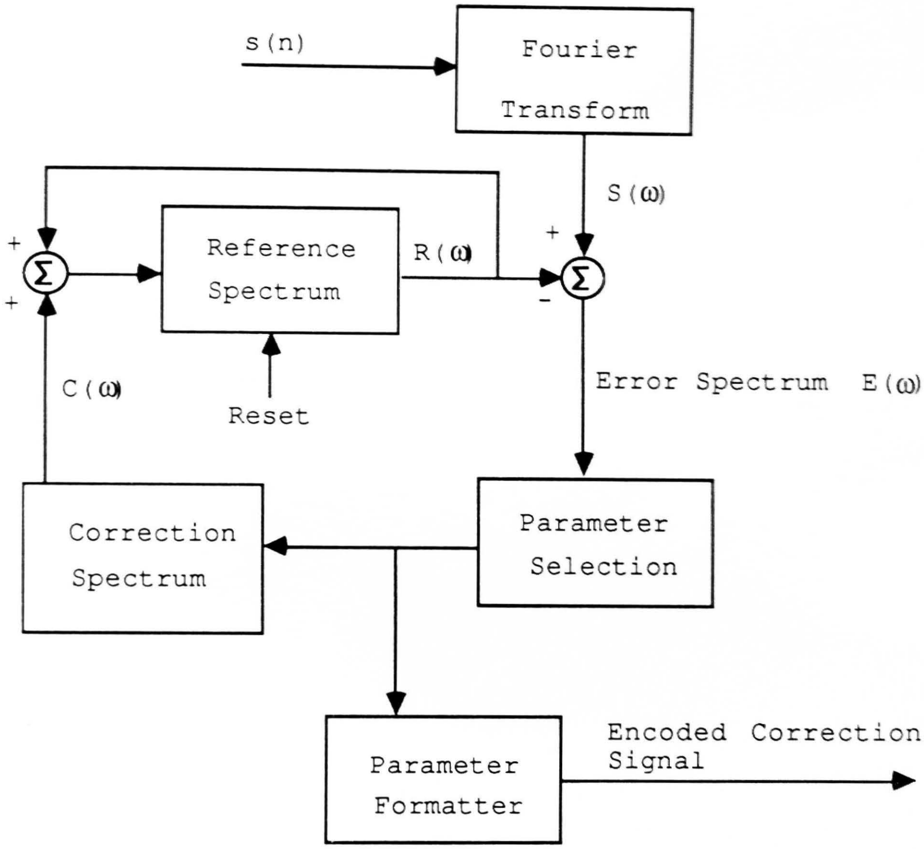
Polar Spectrum Coding (PSC) is a bandwidth-efficient method of encoding analog signals. The heart of the encoder is a frequency domain spectrum waveform builder which uses a minimum of parameters as inputs. Correlation between successive frames of data samples is exploited to reduce the bandwidth needed in transmission. At the receiver, the decoder is a similar system to reconstruct a model of the original spectrum. An inverse DFT operation creates an output waveform signal.

PSC is based on the assumption that the only information that is transmitted is always the information that is most critical for allowing the receiver to match the original waveform as closely as possible. To this end, PSC is a strategy which determines which information is most critical, and then encodes this information in the most bit-efficient manner. To do this, information which is given the name of "corrector(s)" is sent to the receiver. A corrector is a signal which indicates to the receiver the latest changes in the signal. The number of correctors sent during each data sample frame time can range from one to many.

## Polar Spectrum Encoder

A block diagram of a simplified Polar Spectrum Encoder is shown in Figure 9, and Figure 10 shows a flow chart for the Polar Spectrum Encoding algorithm. The analog waveform is suitably sampled and digitally encoded. A low pass filter removes frequency components above the Nyquist frequency for the system sampling rate. The input digital signal, $s(n)$, is divided into time frames (typically 8 ms). A polar Fourier transform is performed on each data frame to obtain the frequency spectrum magnitude and phase $S(\omega)$. The spectrum is compared to a reference spectrum $R(\omega)$, and the difference between the two becomes the error spectrum $E(\omega)$.

The primary task of the encoder is to find the most critical information that must be transmitted across the channel to indicate to the decoder the value of the error spectrum. To do this, parameter values are chosen which give an indication of the most important characteristics of the error spectrum. Parameter values contain information about the magnitude and the phase of the signal spectrum. This information is used to generate a correction signal. The correction signal is essentially another spectrum that, in the ideal case, matches the information in the error spectrum.



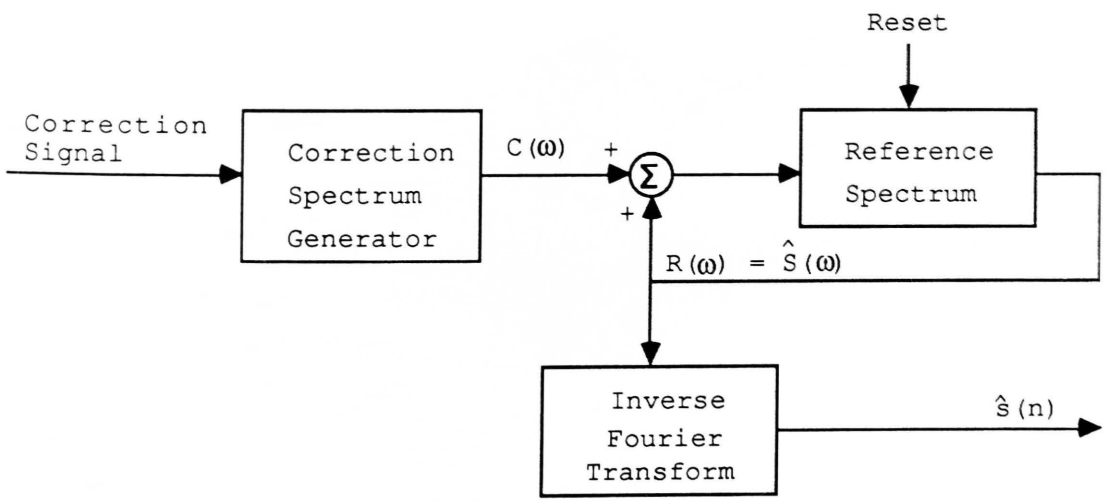




Figure 9. Polar Spectrum System. Encoder and Decoder.

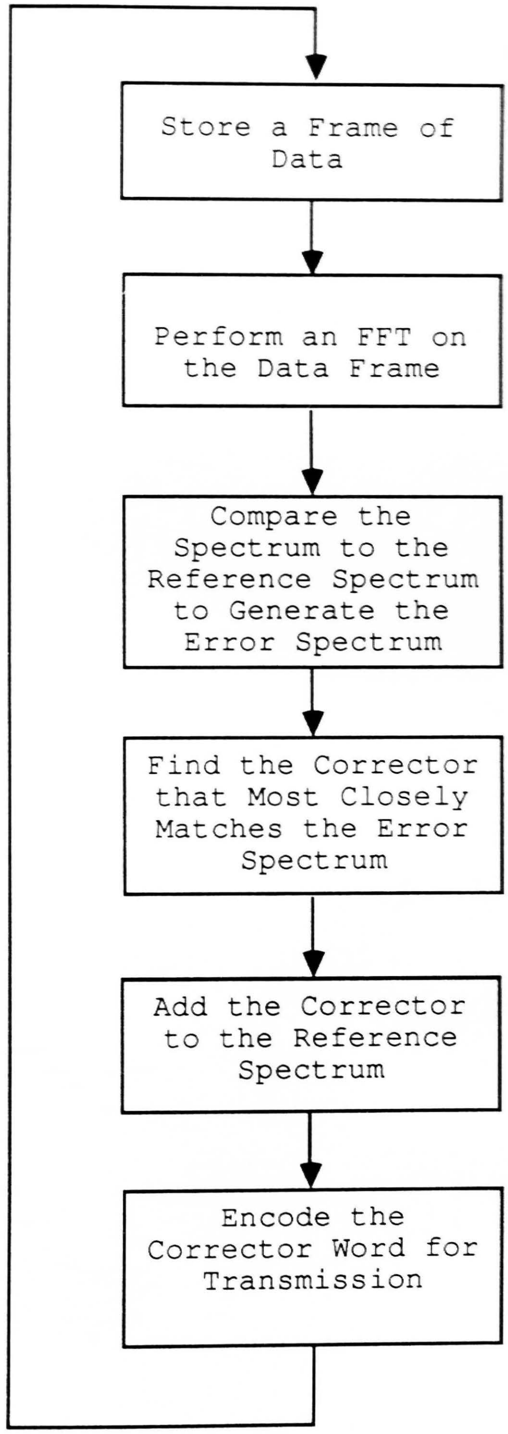



Figure 10. Polar Spectrum Encoding Algorithm Flow Chart.

All of the selected parameters of the corrector (magnitude and phase) are fed to both the parameter formatter and the correction spectrum generator. In the correction spectrum generator, a correction spectrum, $C(\omega)$, is generated which is then added to the value of the reference spectrum to create an updated reference spectrum. In the data formatter, the corrector information is encoded into a correction word $c(n)$ and sent out over the channel. The correction word encoding process simply consists of serializing the bits in the corrector for serial transmission.

The process described above repeats every data frame. At the beginning of a transmission, the reference polar spectrum is reset to an initial value for synchronization with the decoder (possibly flat 0 dB and zero phase for all frequency components). As each frame of data is encoded, the reference spectrum is modified once per frame to track changes in the signal spectrum. PSC transmits magnitude information which causes the receiver to adjust its spectrum estimate to more closely match the spectrum that is present at the transmitter. Gradually, the reference spectrum is shaped to closely match the original spectrum. PSC is an adaptive system. When the speech spectrum is changing slowly over time, very little change is made to the reference spectrum each frame time. Obviously, highly correlated data frames would require less information to be sent since the receiver would not have to change its spectrum appreciably once it has closely matched the original. During times of abrupt changes in the speech spectrum, more pronounced changes in the reference spectrum are made to keep the error spectrum small.

## Polar Spectrum Decoder

Figure 9 shows a block diagram for the Polar Spectrum Decoder. The related algorithm flow chart is presented in Figure 11. Several sections of the decoder and the decoding process are similar or identical to those in the encoder. The correction signal, c(n), is received and used to generate a correction word which represents a correction spectrum. Assuming an error-free channel, the correction spectrum of the decoder matches the correction spectrum in the encoder. Just as in the encoder, the correction spectrum is summed to the reference spectrum during each frame to create a new reference spectrum. At the beginning of the reception of a speech signal, the reference spectrum is reset to match the initial value of the encoder reference spectrum. Therefore, the reference spectrums of the encoder and the decoder are always identical, assuming no channel errors. An inverse Fourier transform is performed on the reference spectrum to reconstruct the time domain speech data frame. The speech data frame is converted to an analog signal, and low pass filtering is performed on the output to remove high frequency spectrum replicas caused by the sampling process.

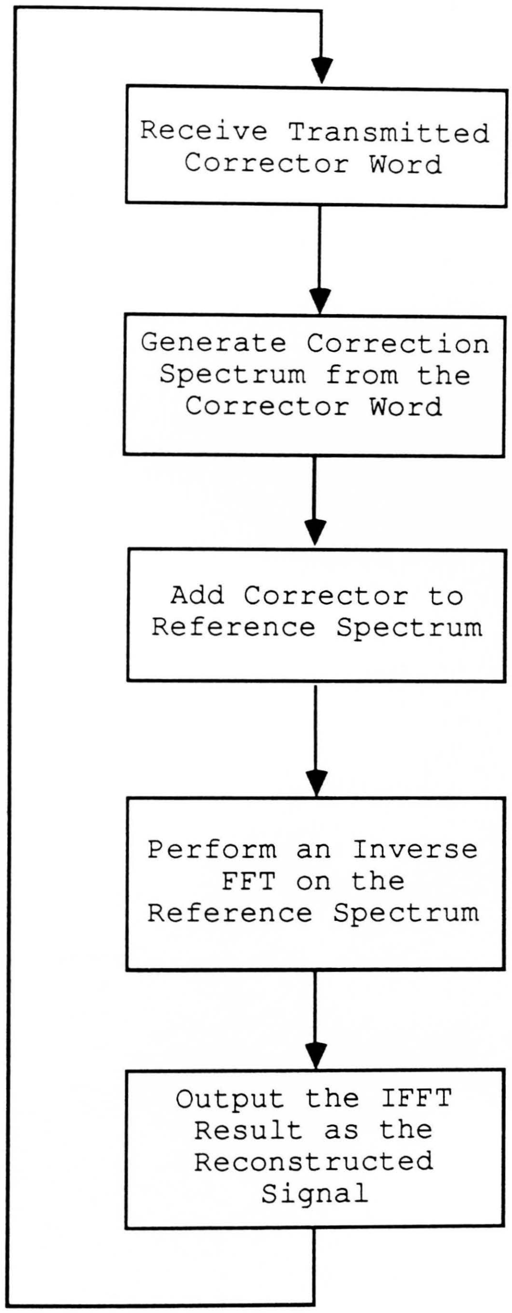



Figure 11. Polar Spectrum Decoding Algorithm Flow Chart.

Magnitude Information Encoding - Possibilities

Because voiced segments of speech signals vary slowly over time, there is a high correlation between successive frames of speech signals when the frames are relatively short. In the frequency domain, this translates to a high correlation between frames of both the magnitude and the phase information. The magnitude information, however, is different from the phase information in one important aspect. That is, the value of the magnitude of a particular frequency component remains unchanged from frame to frame for maximum correlation. This is not true for phase information. The phase value of a particular frequency component is in general very different from frame to frame, even with high correlation. Since speech signals have, in general, high correlation between successive short frames, and magnitude spectrum values tend to remain relatively unchanged from frame to frame as a result, it becomes advantageous to use this fact to differentially encode the magnitude values. This is, in fact, one of the central ideas in PSC.

Magnitude Parameter Selection

Since magnitude information can be transmitted differentially, bandwidth can be conserved compared to a non-differential transmission of information. In order to minimize the bandwidth, and at the same time maximize the quality of the reconstructed speech, parameters of the PSC system must be carefully chosen. The goal is to pack as much critical information as possible into each transmitted data bit.

Definitions. Before details of PSC can be explained, it is important to start with a set of definitions that will simplify later discussions. Some of the terms are used in related literature and are reiterated here to narrow their meaning to the problem at hand. Other terms that are defined here are unique to PSC due to the novelty of PSC.

Signal Frame. A group of successive quantized samples of a speech signal is a signal frame. In most speech processing literature, the number of samples in a frame is designated by N. PSC is performed on one signal frame of a speech signal at a time.

Reference Magnitude Spectrum. The synthesized speech spectrum of each frame is the reference magnitude spectrum. There is one reference magnitude spectrum for each signal frame. PSC uses signal frames to determine the operation that must be performed on the reference magnitude spectrum in each frame.

Error magnitude spectrum. The difference between a signal magnitude spectrum and a reference magnitude spectrum is the error magnitude spectrum. One goal of PSC is to keep the error magnitude spectrum as small as possible.

Corrector. A corrector is a group of data bits that represents the change that will be made to the reference spectrum of the next frame to cause it to be different from the reference spectrum of the present frame. There can be multiple correctors in each frame. In the case of a single corrector per frame, the corrector is made to be as close as possible to the error spectrum. In the case of multiple correctors in each frame, the summation of the correctors in a frame is made to be as close as possible to the error spectrum.

Magnitude Corrector. A magnitude corrector is the magnitude portion of a corrector.

Magnitude Corrector Sign. Each magnitude corrector must either add to or subtract from the reference magnitude spectrum of the present frame to shape the reference magnitude spectrum of the next frame. The magnitude corrector sign determines which operation is performed.

Frequency Bin. When an FFT is performed on a sampled signal, the spectrum (both magnitude and phase) is computed at discrete frequency values. These discrete frequency values are the frequency bins.

Center Frequency. The center frequency is the frequency bin at which the center of the magnitude corrector is positioned. If the magnitude corrector has an even number of frequency bins, then the center frequency is defined as the closest frequency bin below the center of the magnitude corrector.

Start Frequency. The start frequency is the frequency bin at which the lowest frequency bin of the magnitude corrector is positioned.

Width. The width of a magnitude corrector is the number of frequency bins that the magnitude corrector spans, including the frequency bins at both ends of the magnitude corrector. The width is a measure of the bandwidth of the corrector.

Gain. The gain of a magnitude corrector is the magnitude of the magnitude corrector at its maximum magnitude.

Shape. The envelope of the corrector is its shape. For example, a triangular corrector has its highest (or lowest) magnitude at its center frequency.

Examples. An example of a PSC system serves to illustrate some of the concepts presented above. In this example, the shape of the magnitude corrector is always rectangular. Assume an input voice signal has the signal magnitude spectrum shown in Figure 12a; the numbers in the figure are

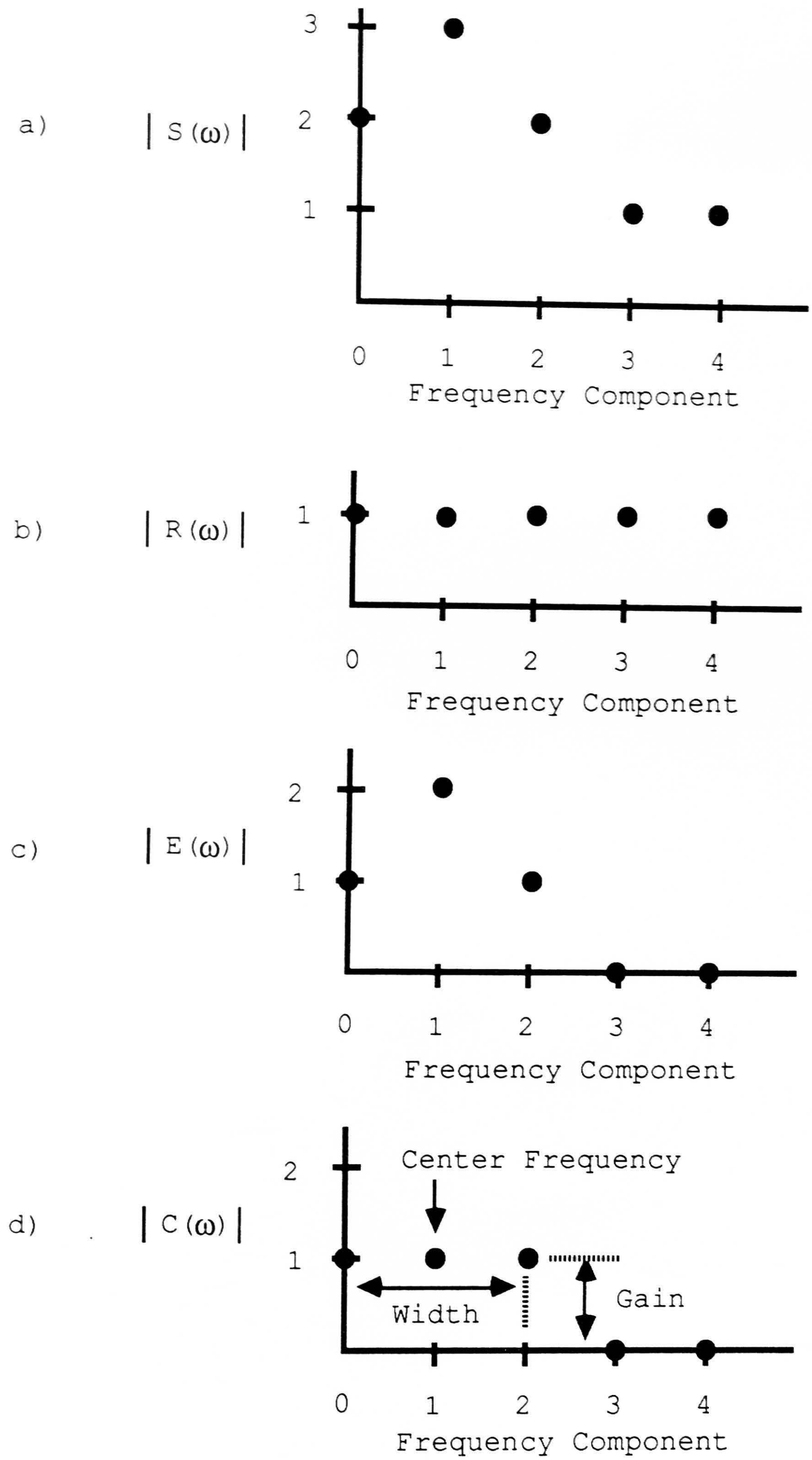



Figure 12. Examples of Magnitude Spectrum Types: a) Signal Spectrum; b) Reference Spectrum; c) Error Spectrum; and d) Corrector Spectrum.

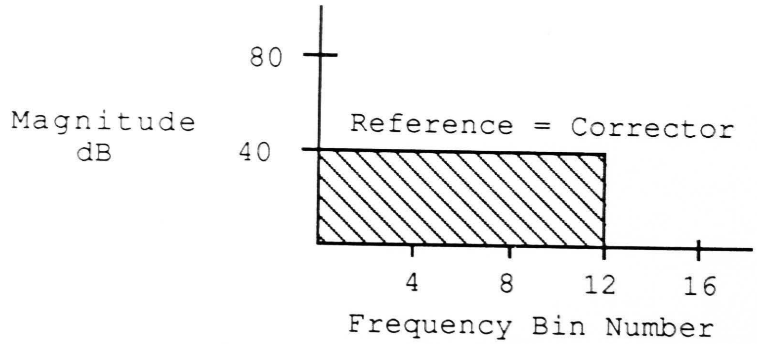
normalized for simplicity. The current reference magnitude spectrum is shown in Figure 12b; it consists of a magnitude of 1 for all frequency components. The error magnitude spectrum, shown in Figure 12c, is the difference between the signal magnitude spectrum and the reference magnitude spectrum. Note that there is no error at frequency components 3 and 4, and the maximum error occurs at frequency component 1. The closest that a rectangular corrector can match the error magnitude spectrum is shown in Figure 12d. The magnitude corrector has a magnitude of 1, a width of 3, and a center frequency of 1. This example corrector would leave a residual error at frequency component 1 with a magnitude of 1.

Another example uses an actual simulation output. Figure 13 shows a listing of the spectrum values for the first three frames of a PSC reference spectrum. The order of the listing is {magnitude 1, phase 1, magnitude 2, phase 2, etc.}. The magnitude is in decibels and the phase is in radians. In this example, there are 64 time domain samples per frame. The time domain samples are real values, so the frequency domain spectrum is odd symmetric. Therefore, only 33 magnitude/phase pairs are shown in each frame. Figure 14 shows a simplification of the shape of the corrector and the reference spectrum in each frame, and Figure 15 gives a printout of the reference spectrum in the three frames.

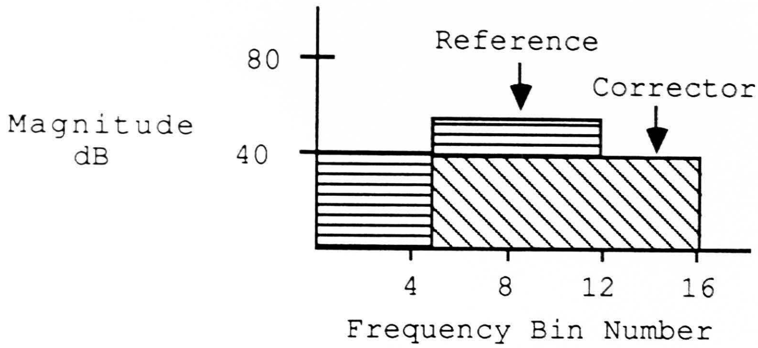
| | | | | |
|---|---|---|---|---|
| 40.00000 | 3.141593 | 40.00000 | -1.963496 | 40.00000 |
| 2.356194 | 40.00000 | 3.141592 | 40.00000 | -4.7683716E-07 |
| 40.00000 | -0.7853985 | 40.00000 | 2.356194 | 40.00000 |
| 2.748893 | 40.00000 | 0.3926985 | 40.00000 | -4.7683716E-07 |
| 40.00000 | 2.356194 | 40.00000 | 2.356194 | 0.0000000E+00 |
| 1.963495 | 0.0000000E+00 | 1.963495 | 0.0000000E+00 | 2.356194 |
| 0.0000000E+00 | 2.356194 | 0.0000000E+00 | 3.0845[illegible]2 | 0.0000000E+00 |
| 0.8777287 | 0.0000000E+00 | -2.592524 | 0.0000000E+00 | 1.517679 |
| 0.0000000E+00 | 2.237066 | 0.0000000E+00 | 2.090573 | 0.0000000E+00 |
| -6.2760592E-02 | 0.0000000E+00 | 0.5984929 | 0.0000000E+00 | 0.2410765 |
| 0.0000000E+00 | 0.4918056 | 0.0000000E+00 | 1.647235 | 0.0000000E+00 |
| 3.060455 | 0.0000000E+00 | 2.646136 | 0.0000000E+00 | 0.6872082 |
| 0.0000000E+00 | 1.609174 | 0.0000000E+00 | 0.8355649 | 0.0000000E+00 |
| 0.5971465 | | | | |
| 40.00000 | 3.141593 | 40.00000 | -2.748894 | 40.00000 |
| 1.963495 | 40.00000 | 2.356194 | 56.56854 | -0.7853985 |
| 56.56854 | -1.570797 | 56.56854 | 0.7853978 | 56.56854 |
| 1.178097 | 56.56854 | -1.178098 | 56.56854 | -1.178098 |
| 56.56854 | 0.7853978 | 56.56854 | 1.963495 | 40.00000 |
| -2.748894 | 40.00000 | 2.748893 | 40.00000 | 1.570796 |
| 40.00000 | 1.963495 | 0.0000000E+00 | -1.018150 | 0.0000000E+00 |
| -1.182255 | 0.0000000E+00 | -0.8708127 | 0.0000000E+00 | 2.783932 |
| 0.0000000E+00 | -0.9019313 | 0.0000000E+00 | [illegible].299529 | 0.0000000E+00 |
| -0.1600463 | 0.0000000E+00 | -2.105021 | 0.0000000E+00 | 1.239836 |
| 0.0000000E+00 | 0.7298379 | 0.0000000E+00 | -0.8207171 | 0.0000000E+00 |
| 0.7351908 | 0.0000000E+00 | 2.179105 | 0.0000000E+00 | 1.150989 |
| 0.0000000E+00 | 2.82[illegible]482 | 0.0000000E+00 | -2.347433 | 0.0000000E+00 |
| 2.761479 | | | | |
| 40.00000 | 3.141593 | 40.00000 | 1.963495 | 56.56854 |
| 1.570796 | 56.56854 | -4.7683716E-07 | 69.28204 | -0.7853985 |
| 69.28204 | 1.570796 | 69.28204 | 0.7853978 | 69.28204 |
| 2.356194 | 69.28204 | -1.570797 | 69.28204 | 1.963495 |
| 69.28204 | 1.963495 | 69.28204 | -1.570797 | 56.56854 |
| -1.570797 | 56.56854 | -0.7853985 | 40.00000 | 0.3926985 |
| 40.00000 | -1.178098 | 0.0000000E+00 | 1.579265 | 0.0000000E+00 |
| 2.153317 | 0.0000000E+00 | -1.798485 | 0.0000000E+00 | -0.9481781 |
| 0.0000000E+00 | -4.3811798E-02 | 0.0000000E+00 | 2.469434 | 0.0000000E+00 |
| -1.956509 | 0.0000000E+00 | -1.669608 | 0.0000000E+00 | -2.862424 |
| 0.0000000E+00 | 1.990621 | 0.0000000E+00 | 1.581330 | 0.0000000E+00 |
| 0.3018646 | 0.0000000E+00 | 1.906259 | 0.0000000E+00 | -0.7262104 |
| 0.0000000E+00 | 5.9993505E-02 | 0.0000000E+00 | 3.073110 | 0.0000000E+00 |
| 1.91953[illegible] | | | | |

Figure 13. Listing of Reference Spectrum for the First Three Frames of a Speech Signal; 33 Pairs of Magnitude and Phase Per Frame.

a)




b)




c)

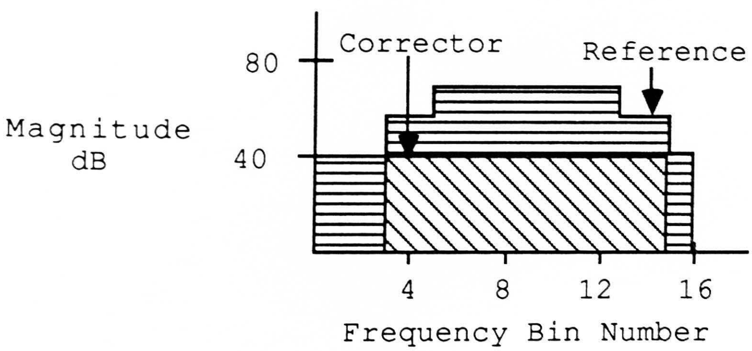



Figure 14. First Three Frames of an Example PSC Output, Showing the Corrector and Reference Spectrums for Each. a) Frame 1; b) Frame 2; c) Frame 3.

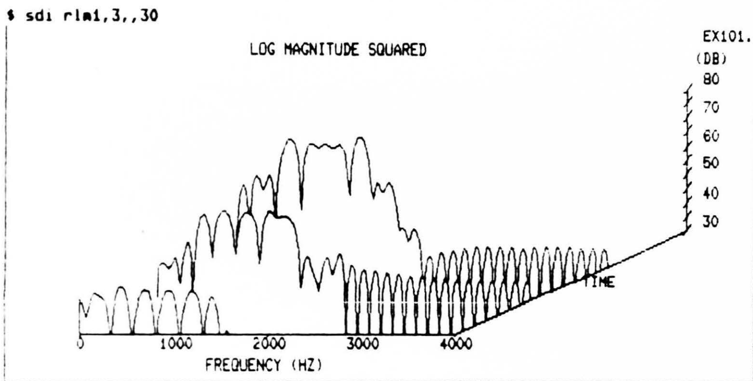



Figure 15. Reference Spectrum of the First Three Frames of a PSC Output.

Independent Magnitude Parameter Selection. Independent magnitude parameter selection allows each parameter of the corrector(s) in each frame to be independently chosen. The width, center frequency, and other parameter choices are selected based on individual algorithms that search for an optimimum value of a particular parameter. Figure 16 shows a flow chart for the independent magnitude parameter selection algorithm. Once the error spectrum is generated, each parameter value is, in turn, optimized. After all parameter values have been chosen, the parameters are used to generate the corrector.

Criteria for Selection of Parameter Values. Careful selection of the values that the parameters are given, as well as the choices of values that are available to the parameters, optimize the performance of a PSC system. One criterion that can be imposed on parameter values is to minimize the mean squared error of the spectrum in each frame. This is a relatively simple constraint to place on parameters after all of the possible choices for values have been designed into the system, since PSC is a frequency domain algorithm. Another method to grade parameter values is to minimize the mean squared error of the time domain waveform. This method requires more computation in the encoder; an inverse FFT of the reference spectrum must be computed before the mean squared error can be calculated.

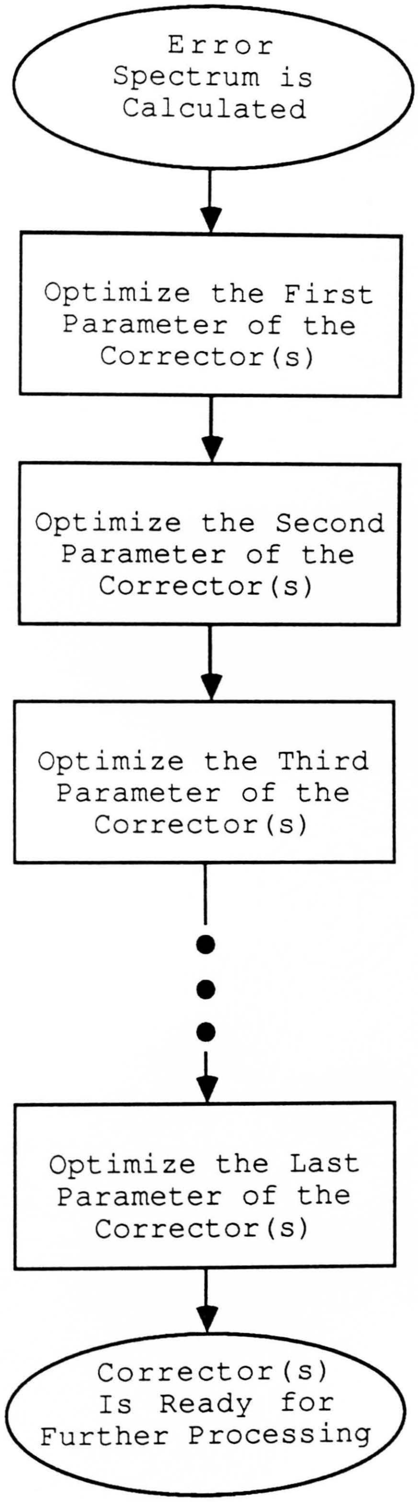



Figure 16. Independent Magnitude Parameter Selection Algorithm.

An intuitively appealing method for determining the range of values that each parameter should have involves experimentally deriving probability density functions for the parameter values. The parameter values can be chosen to approximate a uniform distribution of probability. If the PSC system is tried on a group of sentences with varying characteristics, the probability density functions computed, and parameter values adjusted to change the functions closer to uniform, a set of parameter values that maximize the entropy of the PSC encoded signal will be obtained.

In order to have a coding system which maximizes the quality of speech transmission while minimizing the transmission bit rate, it is necessary to carefully select the parameters of the system. In particular, the following parameters are necessary (but not sufficient) to completely define a PSC system:

1) Speech sampling rate;

2) Speech quantization parameters;

3) Length of data frame;

4) Number of magnitude correctors per frame;

5) Range and resolution of the center frequency for each magnitude corrector;

6) Range and resolution of the width of each magnitude corrector;

7) Range and resolution of the gain of each magnitude corrector; and

8) Shape of the magnitude corrector.

Parameters 1 and 2 are beyond the scope of this paper. However, speech should be sampled at the Nyquist rate or higher to ensure reconstruction of all frequencies of interest. Quantization of analog signals is assumed to be such that quantization noise is negligible and yet dynamic range is sufficient.

The length of the data frame is of critical importance to the success of a PSC system. If the length is chosen to be too long, adjacent speech sample frames will typically be relatively uncorrelated. This puts the system under constant stress trying to reconstruct the original signal at the receiver. An analogous behavior occurs when a DM system experiences slope overload; the reconstructed signal cannot change fast enough to keep pace with the change in the original signal. In addition, the excessive modifications that must be made to the spectrum each frame period might cause discontinuities that are perceived by the human ear. This would make the reconstructed speech sound unnatural. A data frame length that is chosen to be too short will not cause the correlation problems of excessively long frames, but the frequency spectrum resulting from the Fourier transform will not have enough information in it to allow the system to adequately reconstruct the spectrum at the receiver. The resulting frequency distortion would translate into a change in the sound heard.

Parameters 4, 5, 6, 7, and 8 make PSC a unique method of encoding speech signals. A tradeoff must be made in selecting the number of magnitude correctors that are transmitted each frame period. A greater number of correctors allows faster response to rapidly changing waveforms; an infinite number of correctors allows the magnitude of the spectrum to be reconstructed almost perfectly, even when there is no correlation between adjacent data samples. On the other hand, a larger number of magnitude correctors requires more bits to be encoded and therefore requires a higher bit rate. A number of magnitude correctors per data frame between one and five appears to be a reasonable quantity.

The proper amount of resolution given to the center frequency, width, and gain of each corrector is another balance between better performance and a higher bit rate. A finer resolution in these three parameters means that the original speech spectrum can be reconstructed more precisely. However, more bits are required to obtain greater resolution.

A related parameter that must be defined is the range of frequencies, widths, and gains that each magnitude corrector can span. The simplest encoding scheme allows each corrector to have any center frequency, width, or gain within the resolution of the system. Although this gives the system the most flexibility, it increases the bandwidth required for transmission in systems with multiple correctors. Each corrector can be given its own sub-range of frequencies, analogous to the technique of Sub-Band Coding, and widths and gain in order to serve particular needs of spectrum shaping.

For example, two out of three correctors might be given the range of frequencies between 50 Hz and 1000 Hz, where formant frequencies are dominant; and the third corrector might be used to cover the range between 1000 Hz and 3000 Hz, if it is determined that less resources are needed in that area. This allows each corrector to have a narrower resolution than would be possible if all correctors had to span the entire range of frequencies between 50 and 3000 Hz.

As another example, the first of the two 50-1000 Hz correctors might be given a range of gain of between 1 and 3 dB. If the second corrector has a range of from 3 to 12 dB, then the two correctors can be used as "fine" and "coarse" magnitude adjusters, respectively.

Center Frequency. An important thing for a PSC system to do is to efficiently and accurately calculate the correctors. Efficiency is important because noticeable delays, especially in speech conversations, are irritating. After the magnitude error spectrum has been obtained (Figure 10), the next step is to calculate the center frequencies of the correctors. There are different methods in which the center frequencies can be determined. One method picks the frequencies of the error magnitude spectrum with the highest gain. This is reasonable since the error magnitude spectrum peaks represent the frequencies at which the reference magnitude spectrum deviates the most from the signal magnitude spectrum, and one purpose of the encoder is to minimize the deviation. This method requires only a peak picker to determine the center frequency. A second, more computationally-intensive method finds areas under the error magnitude spectrum curve with the greatest values. An attempt is made to find the center frequencies from which magnitude correctors can be generated to most closely match the curve of the error magnitude spectrum. Figure 17 shows an example of an error magnitude spectrum in which each type of algorithm would pick a different center frequency. The first method would pick a center frequency of $f_1$ Hz, since this frequency has the highest peak. However, $f_2$ would be chosen by method two, since this curve has more area under it.

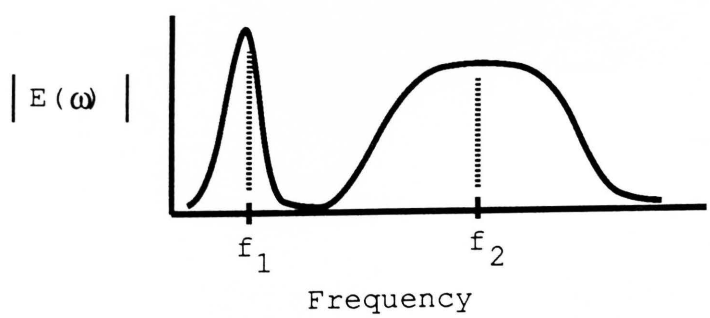



Figure 17. Examples of Center Frequency Selection.

It is hard to generalize which method gives better performance. The first method is quicker and has the advantage that the "loudest" deviations will be dealt with. The second method requires more computation but has the advantage that it selects the maximum energy content of the error magnitude spectrum to correct.

Width. Both the error spectrum and the selected center frequencies are used to calculate the widths of the magnitude correctors in Figure 10. A relatively simple way to select each width is to pick the one that causes the corresponding corrector to most closely match the error spectrum curve. In Figure 18, one width at a time is paired with the center frequency, and a corrector is generated. The difference between the corrector and the error spectrum is calculated, and the result is compared to the best match so far. The "best match" storage location is initialized each frame to a maximum value. Each time a new width results in a lower difference, the difference is placed into the "best match" storage location and the corresponding width value is placed in the "width" storage location. After all widths have been tried, the final width value in the "width" storage location is the one selected for the particular center frequency.

Gain. There are several possible ways to specify gain in a magnitude corrector. One way is to use decibels for the gain values. If decibels are used, then the actual energy value of the gain depends on the values of the reference spectrum. For example, 3dB gain in a corrector added to a 100dB value in a reference spectrum is a much larger energy value than 3dB added to 50dB. Another method of specifying gain is to use linear values instead of logarithmic. This method has the complication that it takes very large numbers to appreciably change the reference spectrum if the values of the reference spectrum to be changed are already very large values.

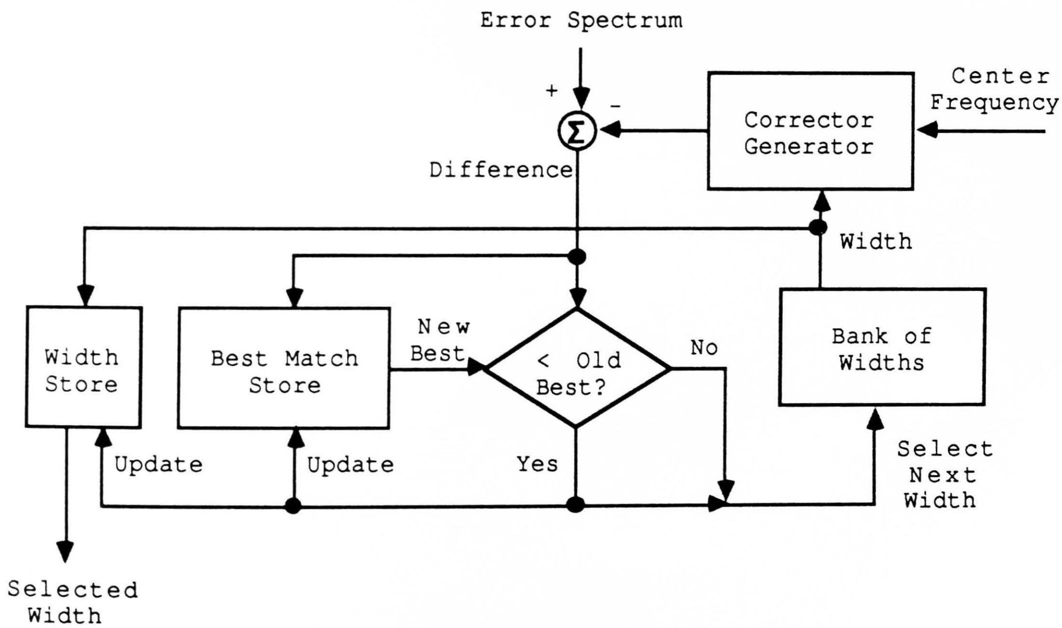



Figure 18. Selection of Width.

Although extra bits can be used to indicate gain, it is possible to imply gain in the correctors without the need of extra bandwidth for transmission. One way to do this is to assign each corrector a fixed gain. Another way is to make the gain adaptive. That is, successive identical correctors increase in gain values. A possibility for very steep steps in gain is to send two or more identical correctors in the same word; the decoder could interpret this as a signal to increase the gain of this set of correctors drastically.

The determination of whether a magnitude corrector has a positive sign or a negative sign is based on whether the part of the error frame which is being minimized has a positive value or a negative value. If the sign is positive, then the magnitude corrector is added to the reference frame. If the sign is negative, then the magnitude corrector is subtracted from the reference frame.

Shape. There are countless options available for the shape of a magnitude corrector. Several examples are rectangular, triangular, and raised cosine. It is possible to have the choice between several shapes in the same PSC system.

Number of Correctors. PSC is not limited to one corrector per frame. In fact, any number of correctors per frame can be transmitted as long as the upper limit on the resultant transmission bit rate is not exceeded.

Example. The overall problem is one of deciding what the choices will be for each corrector. The greater the number of choices given for each corrector, the higher the resultant bit rate will be. An ideal PSC system will minimize the required bit rate by taking advantage of the high correlation of speech signals, while at the same time allowing a quick reaction time to abrupt signal changes. Although the ideal cannot be realized, intelligent tradeoffs result in performance that approaches the ideal.

The following example describes the determination of the transmission bit rate of a PSC system. Assume that there are three magnitude correctors that are generated and transmitted each frame period, which is 20 ms. All three magnitude correctors have three bits to specify the center frequency. Corrector 1 has the following center frequency options (in Hertz): 50, 100, 150, 200, 300, 400, 600, and 800. Corrector 2 has the frequencies of 100, 200, 300, 400, 500, 600, 700, and 800. Corrector 3 has the choices of 1000, 1200, 1400, 1600, 1800, 2000, 2400, and 3000. All three magnitude correctors have 2 bits to specify the width; in all cases, the choices are (in Hertz): 25, 50, 100, 200, 400, 800, 1600, 3000. One bit for each magnitude corrector is used to specify the sign of the corrector. In this example, assume that a gain of 3 dB is implied for each corrector. However, if two identical correctors are sent in a frame or in adjacent frames, then the gain of the second corrector is interpreted to be 6 dB.

Therefore, each magnitude corrector requires 6 bits to specify its sign (1 bit), center frequency (3 bits), and width (2 bits). There are three magnitude correctors that are sent every 20 ms; therefore, a transmission bit rate ($BR_T$) of

$$BR_T = \frac{1\ \text{frame}}{0.02\ \text{s}} \times \frac{6\ \text{bits}}{\text{corrector}} \times \frac{3\ \text{correctors}}{\text{frame}} = 900\ \text{bps}$$

is required for the transmission of the magnitude spectrum information in this example PSC system.

Exhaustive Search Magnitude Parameter Selection. Exhaustive search magnitude parameter selection calculates the error resulting from all possible combinations of parameter values in the magnitude corrector and then selects the combination that minimizes the error. The magnitude corrector combination that most closely matches the error magnitude spectrum is the corrector that minimizes the error. The exhaustive search algorithm is illustrated in Figure 19. As each combination of magnitude corrector parameter values is compared, the difference between the corrector, or group of correctors in the case of multiple correctors per frame, and the error spectrum is calculated. If the resulting difference is the minimum difference calculated so far, then the combination value is stored as the best match so far. After all combinations of parameter values have been tried, the combination stored at the best match is formatted for transmission.

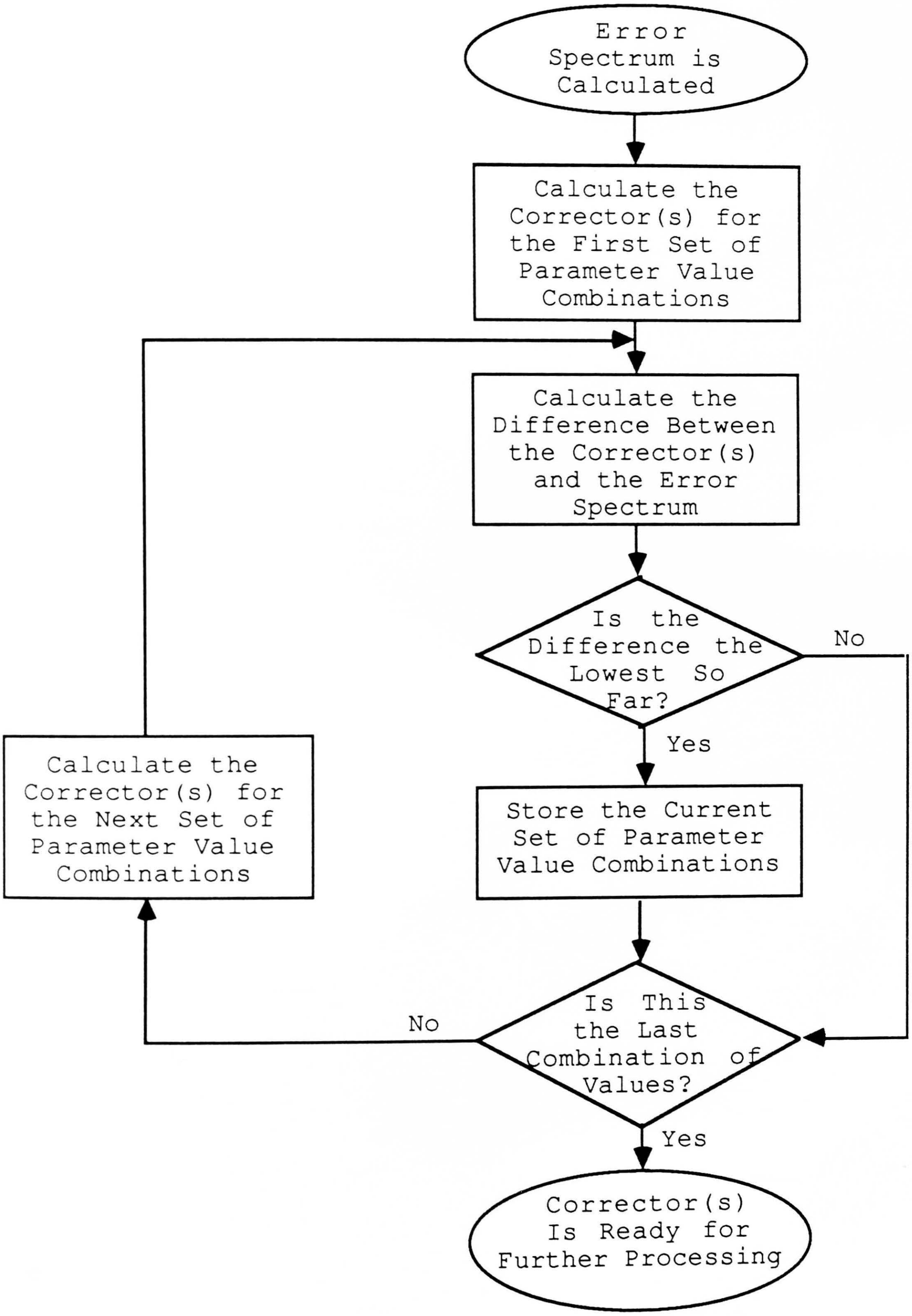



Figure 19. Exhaustive Search Magnitude Parameter Selection Algorithm.

Comparison of Parameter Selection Methods. The exhaustive search parameter selection algorithm is conceptually simpler, but it requires many more combinations of parameter values to be tried. Since the exhaustive search algorithm tries all possible combinations of parameter values, the time required for the algorithm to complete per frame is

$$t_{es} = (N_1 x N_2 x N_3 x \ldots x N_M) x (\text{Time for each combination})$$

where

N1 is the range of values for the first parameter;

N2 is the range of values for the second parameter;

and

$N_M$ is the range of values for the Mth parameter.

The independent selection algorithm chooses each parameter value independently of all other parameters. Therefore, only a subset of all possible parameter value combinations are tested per frame. The time required for the independent selection algorithm to complete each frame, by contrast, is

$$T_{is} = (N_1 + N_2 + N_3 + \ldots + N_M) x (\text{Time for each combination}).$$

Smaller overall errors should result from the exhaustive search algorithm. The exhaustive search algorithm is guaranteed to obtain the best overall combination of parameter values because all combinations are tried. Since the independent selection algorithm does not try all possible combinations of parameter values, the method might find a local minima and overlook the absolute minimum error.

## Phase Information Encoding - Possibilities

The phase information in a spectrum has properties that are different from the magnitude of the spectrum. For example, phase is more important in lower frequency bands where formants are located than in upper frequency bands where the signal has more randomness (Jayant and Noll 1984). Also, phase values tend to be very different from frame to frame, even for highly correlated frames. Because of these differences between the phase and the magnitude of speech spectrums, phase information is encoded in a manner that is different from the magnitude encoding process.

### Phase Parameter Selection

As previously discussed concerning speech signals, during voiced speech there is a high correlation between the phase of a frequency component in two successive frames when the frame length is relatively short. However, the acutal phase value of a frequency component changes drastically from frame to frame. For this reason, phase information is sent in absolute values, not differentially, in a PSC system. As in the magnitude information that is transmitted, phase information must be sent within the constraints set by the parameters of the system.

<u>Definitions.</u> Phase information has less concepts that must be dealt with than magnitude information. Because of this, few definitions are required.

Phase Bin. Each frequency component of a spectrum has an associated phase value. Each frequency component, as it relates to phase, is called a phase bin. An FFT of a signal provides a discrete number of frequency components. The phase value of a particular frequency component is the value of the phase bin.

Lowest Phase Bin. The lowest phase bin is the value of the phase of the lowest frequency component of the spectrum.

Highest Phase Bin. The highest phase bin is the value of the phase of the highest frequency component of the spectrum.

<u>Frequency Range of Phase Information.</u> One parameter of a PSC system is the number of phase bin values that is transmitted each frame. If channel bandwidth were not an issue, then it would be best to send the exact phase value for each phase bin; this would result in the best reconstructed signal. Since infinite bandwidth is a luxury not available to practical communications systems, however, the number of phase bin values that is sent must be minimized.

Algorithms for Frequency Selections. Which phase bins are most critical for the optimization of a PSC system? If certain phase bins are found to be more useful in the reconstruction of the speech waveform, then more of the bandwidth resources must be allocated to these phase bins. Conversely, channel bandwidth must not be wasted on phase bins that do not appreciably enhance the quality of the speech. Two algorithms for the determination of the most critical phase bins are presented below.

Lowest Frequency Components Algorithm. In the lowest frequency components algorithm, the values of the phases of only the lowest frequency components are transmitted. The rationale behind this approach is that formants are located in the lower frequency components. The formants have been shown to be more important to the perceptual quality of the speech signal than some of the other speech waveform components (Jayant and Noll 1984, 487). The lowest frequency components algorithm, therefore, preserves phase information in the portion of the spectrum where critical information is contained.

Greatest Magnitude Algorithm. The greatest magnitude algorithm transmits phase information for the frequency components that have the greatest magnitude in the spectrum. The rationale of this algorithm is that the frequency components with the greatest magnitude are the ones which contain the most energy and, therefore, are the most important to the perceptual quality of the speech. Since this method will preserve phase for frequency components with the greatest magnitude, even if these frequency components are relatively high frequency where phase in not as important (Jayant and Noll 1984, 487), a hybrid approach can be taken in which the phases of frequency components with the greatest magnitude within a certain upper bound of frequency are transmitted.

Phase Quantization. As with many other parameters in PSC, an intelligent tradeoff must be made between resolution and transmission bit rate. How much granularity in phase information can be tolerated? Figure 20 gives some insight into the answer. Figure 20a is an original speech sample; Figures 20b and 20c are waveforms that result when the phase of all frequency bins are quantized into eight and four values, respectively. The eight-level case shows visible degradation, and the four-level case shows serious degradation.

Default Phase Values. In many narrowband communication scenarios, it requires too much channel bandwidth to transmit phase values for every phase bin. Therefore, a decision must be made concerning the phase values of those frequency components that do not have their values

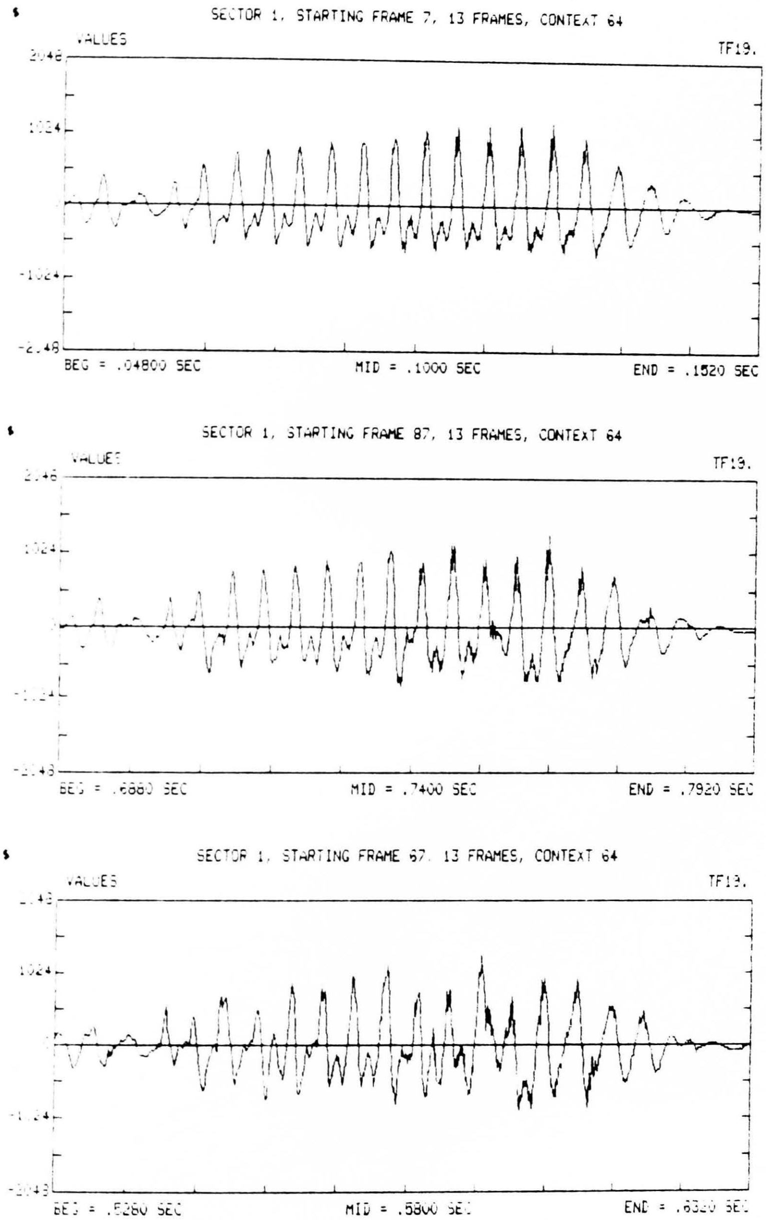



Figure 20. Effects of Phase Quantization on a Speech Waveform. Top: Original. Center: All Phases Quantized to Eight Levels. Bottom: All Phases Quantized to Four Levels.

transmitted. What should these values be? Should the values be time-invariant?

It is possible to allow the phase bins to have locally generated values, instead of transmitting the values. For example, the phases of all phase bins can be set to zero or to random values. Although this significantly reduces the transmission bit rate requirement of a PSC system, the signal is seriously degraded.

Consider the experimentally derived waveforms shown in Figure 21. The first waveform is part of an original sampled signal of the word "with." Figure 21b shows the waveform when the phases of all frequency components are given random values. Note that much of the information in the signal has been lost. The signal no longer has an easily visually discernible pitch; however, a listener will still hear the pitch of the signal. The audio degradation that occurs when the phases are randomized take the form of "buzzing" and "popping" noises that distract from the speech information. The noise is a result of the phase discontinuities that are imposed on the signal at the frame boundaries.

The result of setting the phase of all frequency components equal to zero in each frame is shown in Figure 21c. Drastic signal degradation is apparent in this example. Almost all pitch information is lost. The

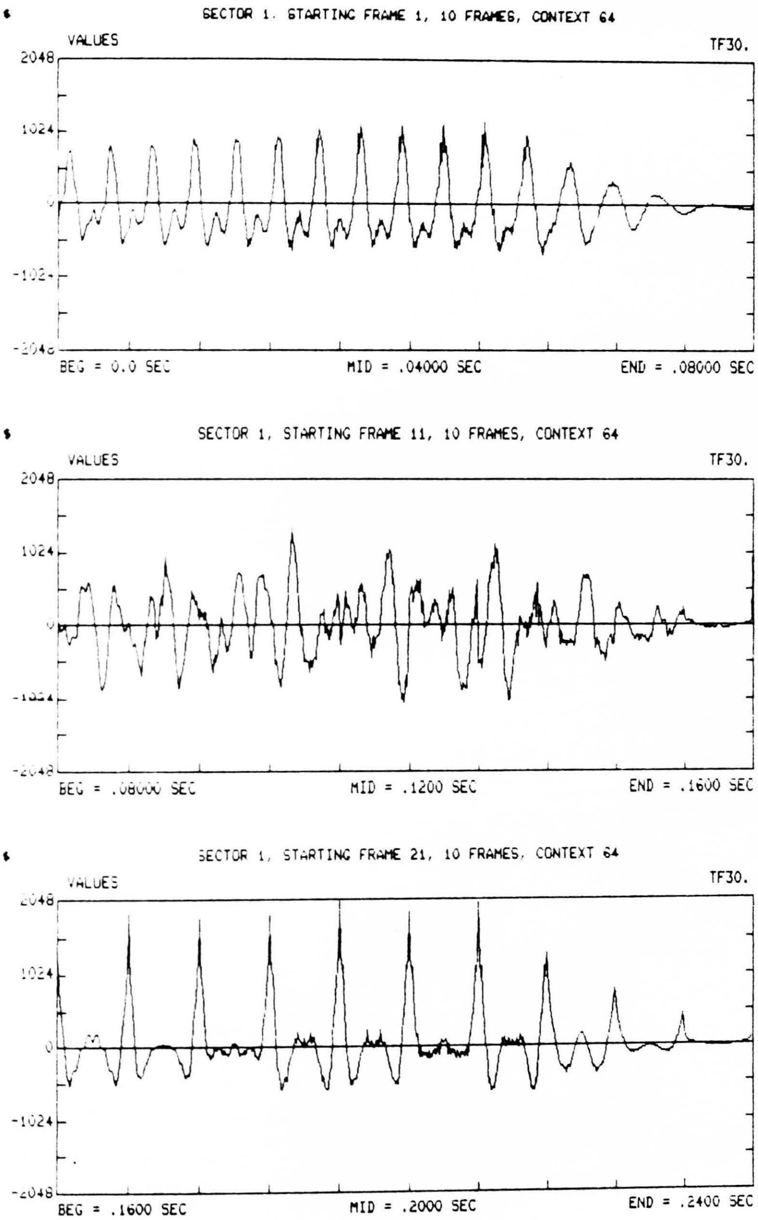



Figure 21. Effects of Different Default Values on a Speech Signal. Top: Original Signal. Center: All Phases Set to Random Values. Bottom: All Phases Set to Zero.

resulting pitch is the frame rate of the signal. The audio distortion in this signal is a monopitch voice signal with a synthetic, "robotic" voice quality. The reason that the pitch is the frame rate can be seen in Figure 22. In Figure 22a, the first three frequency components are assumed to have equal energy and are displayed as three separate sine wave segments. Each sine wave starts every frame boundary at zero phase; therefore, all three sine waves begin the frame in an increasing direction. The summation of these three frequency components is shown in Figure 22b. The summation of all frequency components of the spectrum in a frame will obviously result in a spike at each frame boundary similar to those shown in Figure 21c.

Based on the above observations, it appears that random default values are a better choice than any static value. The problem then becomes one of deciding which phase bins will be quantized to values close to their original values, and which phase bins will be given default values.

## Probability Density Functions

The range of values that each parameter can be assigned in a PSC system must be carefully chosen to maximize the performance. A poor design could result in only one value of a parameter being repetitively chosen.

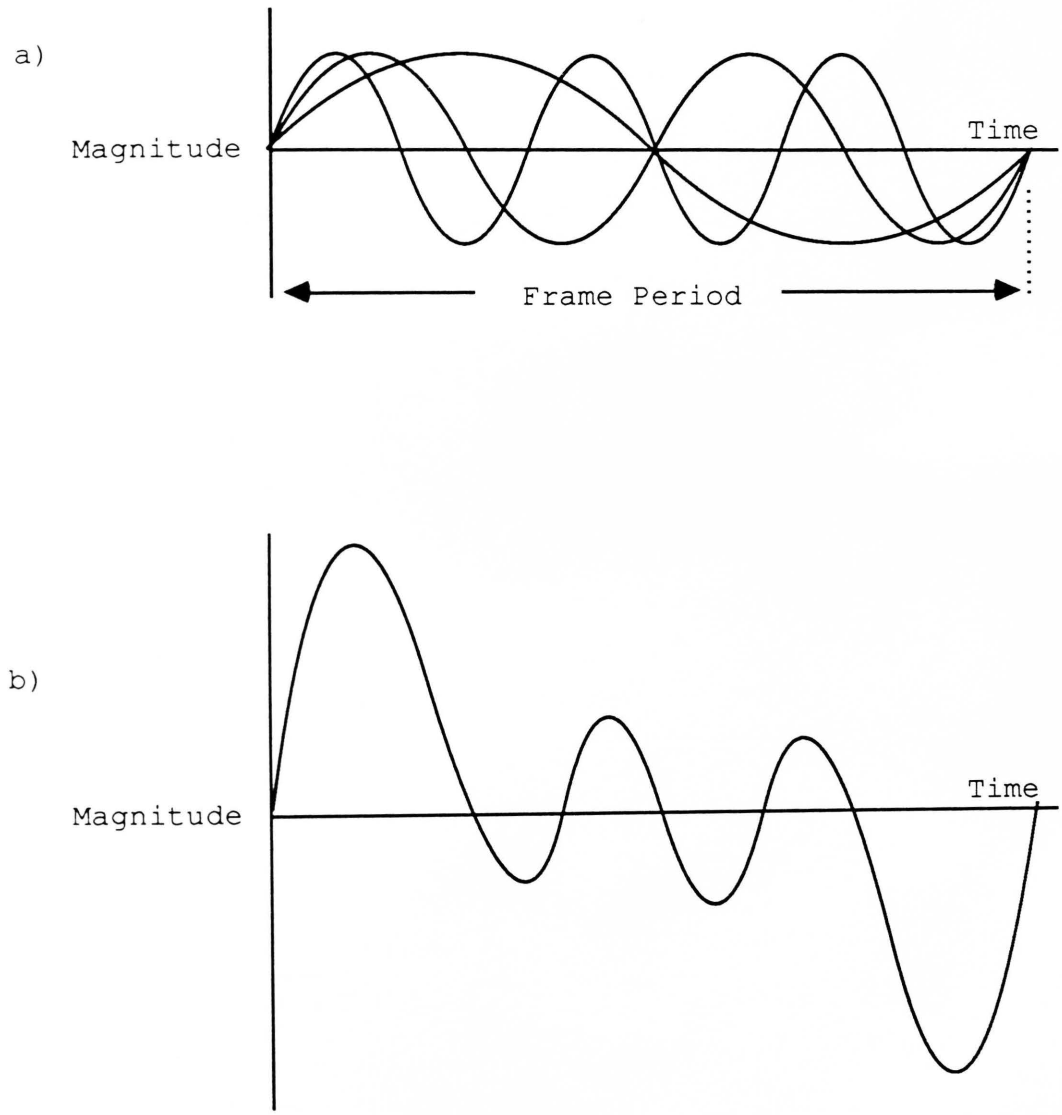



Figure 22. Summation of Frequency Components When Phase Equals Zero for All Frequency Components.

For example, if the possible choices for gain are 100, 200, 300, and 400, but the system generally requires gain values between 10 and 20, then the value of 100 will always be chosen. The extra bandwidth required to encode four choices will be wasted; worse yet, the granularity noise for this poor design will be unacceptable.

Phase parameters are not subject to the range problem of magnitude parameters. The possible values for phase in each frequency component spans the entire range of possible phase values. Therefore, the problem of phase value ranges becomes one of deciding the performance level of the system, not one of determining what the range of values should be.

## Corrector Formatting

Once the correctors have been determined in each frame of the speech signal, they must be formatted into bits for transmission across the communications channel. It is only necessary to transmit one unique word per frame for each unique combination of corrector parameter values. Tables 1 and 2 show an example of the organization of a corrector word. In this example, one corrector per frame is transmitted, and the phases of the first three frequency bins are quantized each frame. As Table 2 shows, there are sixteen choices for center frequency, eight choices for width, and eight choices for gain.

There are two choices for magnitude sign (positive and negative). Table 3 shows that each phase of the first three frequency bins has eight choices. The tables show the number of bits required for each corrector parameter. For the typical parameter values given, typical bit assignments for each parameter value are listed. If the corrector word is formatted in the sequence of {center frequency; width; gain; sign; phase 1; phase 2; phase 3}, the resulting correction word would be "1001011010010001111 1."

TABLE 2

FORMATTING OF THE MAGNITUDE PORTION OF A CORRECTOR WORD.

| | CENTER FREQUENCY | WIDTH | GAIN | SIGN |
|---|---|---|---|---|
| Number of Choices | 16 | 8 | 8 | 2 |
| Number of Bits | 4 | 3 | 3 | 1 |
| Typical Value | 1000 Hz | 200 Hz | 3 dB | Negative |
| Typical Bit Value | 1001 | 011 | 010 | 0 |

TABLE 3

FORMATTING OF THE PHASE PORTION OF A CORRECTOR WORD.

| | PHASE 1 | PHASE 2 | PHASE 3 |
|---|---|---|---|
| Number of Choices | 8 | 8 | 8 |
| Number of Bits | 3 | 3 | 3 |
| Typical Value | $\pi$ | $3\pi/4$ | $-\pi/4$ |
| Typical Bit Value | 100 | 011 | 111 |

## Frame Allocation

Many possibilities exist in the allocation of bit resources in each frame. There are three methods that can be used to allocate bits to represent information. These methods are described below in order of increasing complexity: static, frame independent allocation; static, frame dependent allocation; and dynamic allocation.

In static, frame independent bit allocation, the corrector word follows a fixed, prescribed format every frame. Each parameter is given a fixed number of bits for its value, and each parameter is updated every frame. Static, frame independent bit allocation requires a minimum of hardware to implement the formatting of the corrector word.

Static, frame dependent bit allocation allows different frames to update different parameters. The corrector parameters which are to be updated in each frame depend on the frame number. This could be useful, for example, if a PSC system needs to have phase information updated more often than magnitude information. In this hypothetical case, certain frames in a set can be assigned in which both magnitude and phase is updated, and other frames in the set can update only phase information. The set of frames remains constant, so that bit allocation in a frame set does not vary.

Dynamic allocation of bit resources in frames is similar in technique to ATC (Papamichalis 1987, 178), and it is the most complex of the algorithms. In dynamic allocation, the speech encoder requires some type of signal properties extraction algorithm to determine which parameters require updating. The bits in the frame are then primarily assigned to the parameter that needs to be updated. Additional overhead bits are required to indicate to the decoder which parameters are to be updated in each frame. The advantage of this method is that the bit resources can be utilized where they are needed most.

### Performance in the Presence of Noise

As in all digital communications systems, low sensitivity to noise is desirable. Unfortunately, bit errors can drastically change the action taken by the Polar Spectrum Decoder, especially as it pertains to the magnitude of the spectrum. Errors in the phase of the spectrum are relatively unimportant. Since phase information is not transmitted in a differential manner, any errors in the decoder reference phase spectrum will only affect the speech signal in one frame. Errors in the magnitude of the spectrum, however, can have major effects on the quality of the speech. This is true because magnitude information is primarily differential, as has been discussed. Any error in the magnitude of the reference spectrum in the decoder will cause an error to propogate indefinitely. For example, if an error causes the decoder to set the magnitude of its reference spectrum at 1000 Hz to 60dB instead of 50dB, there will be a 10dB offset at that frequency from that time on, unless the error is corrected. The result will be that the 1000 Hz frequency component will be heard at a level that is 10dB higher than the level of the 1000 Hz frequency component in the origianl speech signal.

There are two basic steps that can be taken to improve the performance of the system in the presence of channel noise. The first thing that can be done to reduce the problems resulting from noise is to reduce the noise itself. A second course of action is to make the PSC system less sensitive to bit errors.

One way to reduce the noise that a PSC system is subjected to is to incorporate a channel coder, such as a block coder or a Viterbi coder, into the communication system. The channel encoder is positioned after the Polar Spectrum Encoder, and the channel decoder is positioned before the Polar Spectrum Decoder.

A way to make a PSC system less sensitive to bit errors is to frequently (perhaps during quiet speech) reset the reference magnitude spectrums in both the encoder and decoder simultaneously. This starts both reference magnitude spectrums over again with a "clean slate" so that errors are not propagated indefinitely. This has the advantage that no extra information is required to be sent to accomplish the synchronization process, except for the message to reset the reference magnitude spectrum.

Another possibility is to frequently send reference signals that are absolute in nature instead of relative. Every few frames, some information is transmitted that indicates to the decoder the exact value to which some portion of its reference magnitude spectrum should be set. For example, a signal might be sent which tells the decoder that its reference magnitude spectrum should be "82 dB at 1000 Hz." If the reference spectrum in the decoder is already set at 82 dB at 1000 Hz, then no corrective action is necessary. On the other hand, any discrepancy can be appropriately dealt with; in this way, the decoder magnitude reference spectrum is gradually modified until it once again recovers from previous errors. The disadvantage of this error-reduction technique is that absolute values require more information to be transmitted than relative values, thus increasing the bandwidth required for speech communication.

A third method to reduce the sensitivity of a PSC system to bit errors is to introduce "leakage" into the corrector values, as is done with CVSD modulation (Papamichalis 1987, 46). In this method, the portions of the magnitude spectrum that are not currently being modified by correctors gradually decrease in value. The effects of transmission errors eventually disappear. The proper amount of leakage will minimize the effects of transmission errors relatively quickly but will not overly stress the system to constantly modify the spectrum to compensate for the leakage.

## PSC Simulation Method

With all of the choices for a PSC implementation that were previously detailed, time prevents experimentation with all of the possible variations. Therefore, certain parameters of a PSC system are assumed in the simulation of PSC. The results that are obtained allow reasonable extrapolations to be made concerning some of the other dimensions that can be explored in other PSC implementations.

In all of the experiments, speech waveforms are lowpass filtered at 4000 Hz and sampled at 8000 samples per second. Eight bits linear quantization is performed on each sample. Background noise is kept at least 40 dB below the maximum value of the speech signal.

There is no transmitter or receiver in the simulation. Instead, the speech is encoded and then decoded. This has two effects on the simulation method. First, there is no signal channel in which errors can be introduced; in order to simulate the effects of channel noise, errors must be forced on the parameters in a random manner. Second, there is no need to format the parameters for transmission. The formatter portion of a PSC system is ignored in the experiments.

Except where otherwise specified, all simulation results are based on a rectangular magnitude corrector shape. This is done primarily for convenience. However, some preliminary work was also performed with a triangular shape, with no noticeable difference in performance.

It is easier to talk about discrete frequency components in terms of bins, and number the bins consecutively. For example, a start frequency of 1 and a width of 3 means that the corrector starts at the first frequency component and extends over three discrete frequencies. Most of the results will be discussed in these terms.

Since a rectangular magnitude corrector is used for most of the experiments, it is easier to specify a "start frequency" instead of a "center frequency." The corrector starts on the start frequency bin and ends on the frequency bin derived by the summation of the start frequency and the width. This change has no effect on the performance of PSC.

The exhaustive search magnitude parameter selection is used in all tests. This method is chosen because it provides potentially better results and is easier to simulate. It is probably the method of choice in a hardware implementation as well.

The choice of start frequency selection is based on a modified form of the method in which areas under the error spectrum curve are minimized. The magnitude of the spectrum is given in decibels. In the calculation of the area under the error spectrum, it is desirable to give more weight to an error that occurs in the frequency bins that contain the most energy. This is true because the frequency bins with the most energy are the loudest and most noticed by the listener. Therefore, to place added weight to the signal spectrum in the frequencies with more energy, the magnitude of the spectrum is squared before PSC processing takes place. This has the effect of increasing the error in the regions of high energy so that PSC will tend to correct the high energy magnitudes. To reverse the distortion due to the squaring process, the square root of the magnitude of the reference spectrum is taken at the end of the PSC algorithm.

Gain is specified by specific number values. An example of a gain value is "1600." Gain is always specified to mean the increase or decrease of the spectrum magnitude; the magnitude is in units of $(\mathrm{dB})^2$. Gain is always explicitly specified, as opposed to a system in which gain is implied but not transmitted.

Frequency selections for phase quantizations are made with the "lowest frequency components" algorithm. Since formants generally reside in lower frequency bins, this method should produce good results. It also saves channel bandwidth since the frequencies that are to be quantized are known to the decoder. The "greatest magnitude" algorithm was tried but did not seem to deliver performance that was worth the resultant higher bit rate.

Default phase values are given random values. Good results are obtained with this default method.

Frame allocation is based on static, frame independent allocation. This method is the easiest to simulate, but it also seems to provide better performance than more complex schemes that were tried. Several experiments were run with static, frame dependent allocation, but the results were not encouraging.

## Simulation Tools

All simulation work was performed on a VAX 11/780. The programs to perform PSC and related testing were written in FORTRAN 77. Interactive Laboratory System (ILS) software was used for Fourier transforms and displays of time and frequency domain waveforms. DEC-LPA microcode was used for D/A and A/D conversions.

Appendix 1 gives an overview of the simulation software. A user's guide for PSC simulation is contained in Appendix 2; several performance examples are given in Appendix 3. The PSC simulation program, listed in Appendix 4, performs PSC on a segment of speech under the user's control; Appendix 5 shows the SNR measurement program used in this research. Translation files to allow speech files to be passed between ILS and ASCII are contained in Appendix 6.

# CHAPTER 4

# RESULTS AND CONCLUSIONS

## Experimentally Derived Probability Density Functions

In order to determine the range of values that should be given to each magnitude parameter, probability density functions were computed experimentally. The number of times the algorithm chose each possible value was recorded. This data allowed intelligent choices to be made concerning the values allowed for each magnitude parameter. The original speech waveform used for a test case is shown in Figure 23. Table 4 shows the range of values allowed for gain, width, and start frequency. In Figure 24, probability density functions are shown for gain, width, and start frequency. The probability density function for the width parameter has a negative slope, but the probability density function for the start frequency parameter is complex. The gain parameter probability density function has a decaying exponential shape. There is more redundancy in the gain parameter, but the higher gain values allow the reference spectrum to be quickly corrected when the signal varies quickly.

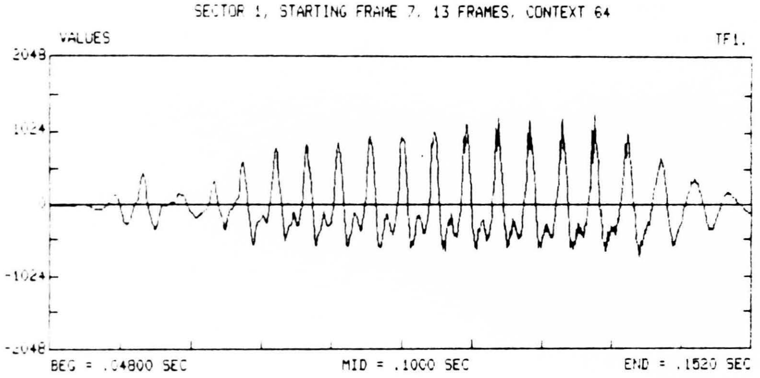



Figure 23. Original Speech Segment Used for Test Purposes. Female Speaker, the Word "With." Thirteen Frames of 64 Samples Each, 8000 Samples Per Second.

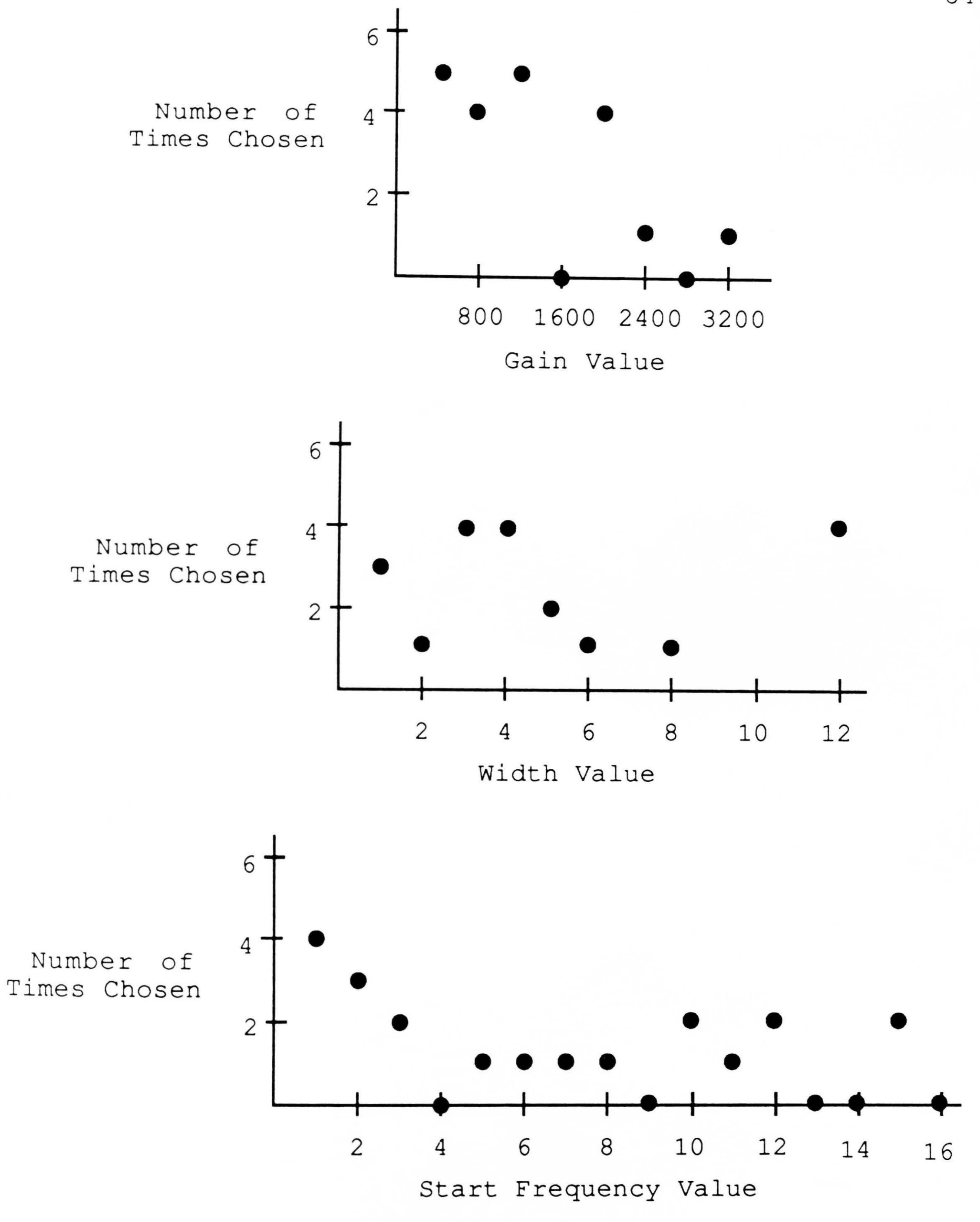



Figure 24. Experimentally Derived Probability Density Functions for Gain, Width, and Start Frequency Values on 20 Frames (160 ms) of Speech.

TABLE 4

RANGE OF VALUES FOR THE MAGNITUDE PARAMETERS FOR THE TEST CASE OF EIGHT VALUES FOR GAIN, EIGHT VALUES FOR THE WIDTH, AND SIXTEEN VALUES FOR START FREQUENCY.

| VALUE | GAIN $dB^{1/2}$ | WIDTH | START FREQUENCY |
|---|---|---|---|
| 1 | 400 | 1 | 1 |
| 2 | 800 | 2 | 2 |
| 3 | 1200 | 3 | 3 |
| 4 | 1600 | 4 | 4 |
| 5 | 2000 | 5 | 5 |
| 6 | 2400 | 6 | 6 |
| 7 | 2800 | 8 | 7 |
| 8 | 3200 | 12 | 8 |
| 9 | - | - | 9 |
| 10 | - | - | 10 |
| 11 | - | - | 11 |
| 12 | - | - | 12 |
| 13 | - | - | 13 |
| 14 | - | - | 14 |
| 15 | - | - | 15 |
| 16 | - | - | 16 |

## Objective Measurements of Parameter Sensitivities

In order to obtain objective measurements of the sensitivity of the performance to various parameters in a PSC coder, a baseline system was chosen against which others could be compared. The chosen baseline has the following characteristics, which have been previously discussed:

- Excitation is a low-pitched female voice
- Speech segment shown in Figure 20
- Low pass filter at 4000 Hz
- Eight-bit samples at 8000 samples per second
- 64 samples per frame
- Rectangular corrector
- One corrector per frame

- Exhaustive search magnitude parameter selection
- Center frequency based on modified area method
- Squared spectrum magnitude for computation
- "Lowest frequency" method phase quantization.

Table 5 gives other characteristics of the baseline system.

TABLE 5

BASELINE PSC SYSTEM CHARACTERISTICS. NUMBER OF POSSIBLE VALUES EACH PARAMETER CAN HAVE, AND THE NUMBER OF BITS REQUIRED PER FRAME.

| PARAMETER | NUMBER OF POSSIBLE VALUES | NUMBER OF BITS REQUIRED |
|---|---|---|
| Gain | 8 | 3 |
| Start freqency | 16 | 4 |
| Width | 8 | 3 |
| Corrector sign | 2 | 1 |
| Phase (bin 1) | 2 | 1 |
| Phase (other bins) | 16 | 4 each |
| No. bins quantized | 16 | 61 total |
| Number of bits required per frame | | 72 |

The number of frames per second is 125:

$$8000 \text{ samples/second} \div 64 \text{ samples/frame} = 125 \text{ fps.}$$

With 72 bits per frame transmitted, the bit rate is 9000 bps:

$$72 \text{ bits/frame} \times 125 \text{ frames/second} = 9000 \text{ bps.}$$

## Signal to Noise Ratio versus Segmental SNR

The SNR is a better measure of the performance of PSC than the segmental SNR. In experiments using both measures, the SNR measure tracked the perceptual quality of the speech much better than did the segmental SNR. This is the opposite of the result that is expected based on literature that is commonly available. Two possible explanations for this occurrence are presented here. First, the measures were taken on speech segments that had no silence intervals, because this is not as important as the performance during non-silence intervals. It is in the silence segments that the segmental SNR gives superior performance over the SNR. Second, the segmental SNR weights the importance of all intervals equally. Therefore, it does not place a greater emphasis on the louder portions of the speech, in which it is more important to get an accurate reconstruction of the speech. The result of these two characteristics of segmental SNR is that the segmental SNR tends to be fairly constant over a wide range of parameter values and ranges. The SNR, on the other hand, tends to track the various parameter combinations in an intuitively pleasing manner. Therefore, the SNR measure is the one used on PSC.

## Corrector Gain

Figure 25 shows the effects of various numbers of gain values on the SNR for a female speaker. As the range of values from which the corrector can choose increases, the SNR tends to increase. However, the increase is not as rapid as with some of the other parameters. This is

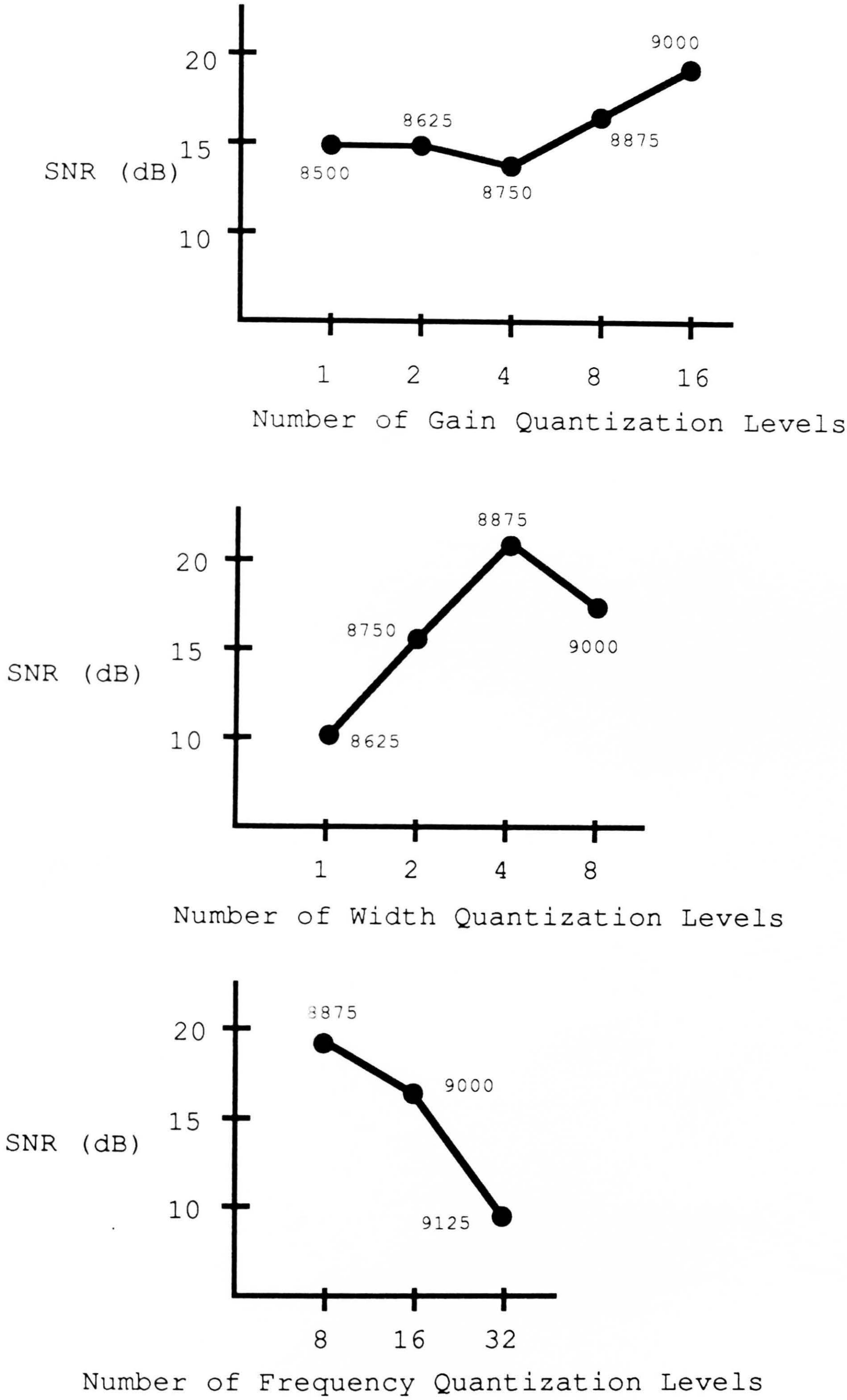



Figure 25. Effects of Gain, Width, and Start Frequency on the Signal to Noise Ratio. Female Speaker. Bit Rates Shown in Small Type.

true for a male speaker also, as seen in Figure 26. Gain is therefore seen to be a relatively insensitive parameter in a PSC system.

## Corrector Width

The effects of various numbers of width values on the SNR for a female speaker are seen in Figure 25. In general, the SNR rapidly increases with an increase in the range of width values in the PSC system. The effects of width on a male speaker are shown in Figure 26. Width is a relatively sensitive parameter.

## Corrector Start Frequency

An unusual result is obtained by varying the number of start frequency values. As Figure 25 shows, the SNR decreases as the number of start frequency choices increases. A close examination of the particular speech segment used in the experiment, shown in Figure 23, helps explains this anomaly. The waveform has primarily low-frequency components. Therefore, it can be best defined if the range of start frequencies is contained in the lower frequency bands. Also, formants are located in the lower frequency bands. It is important that a PSC system spend more channel bandwidth resources correctly modeling the formants than correctly synthesizing higher frequencies. The algorithm often chooses higher frequency components if they are available, and this causes a

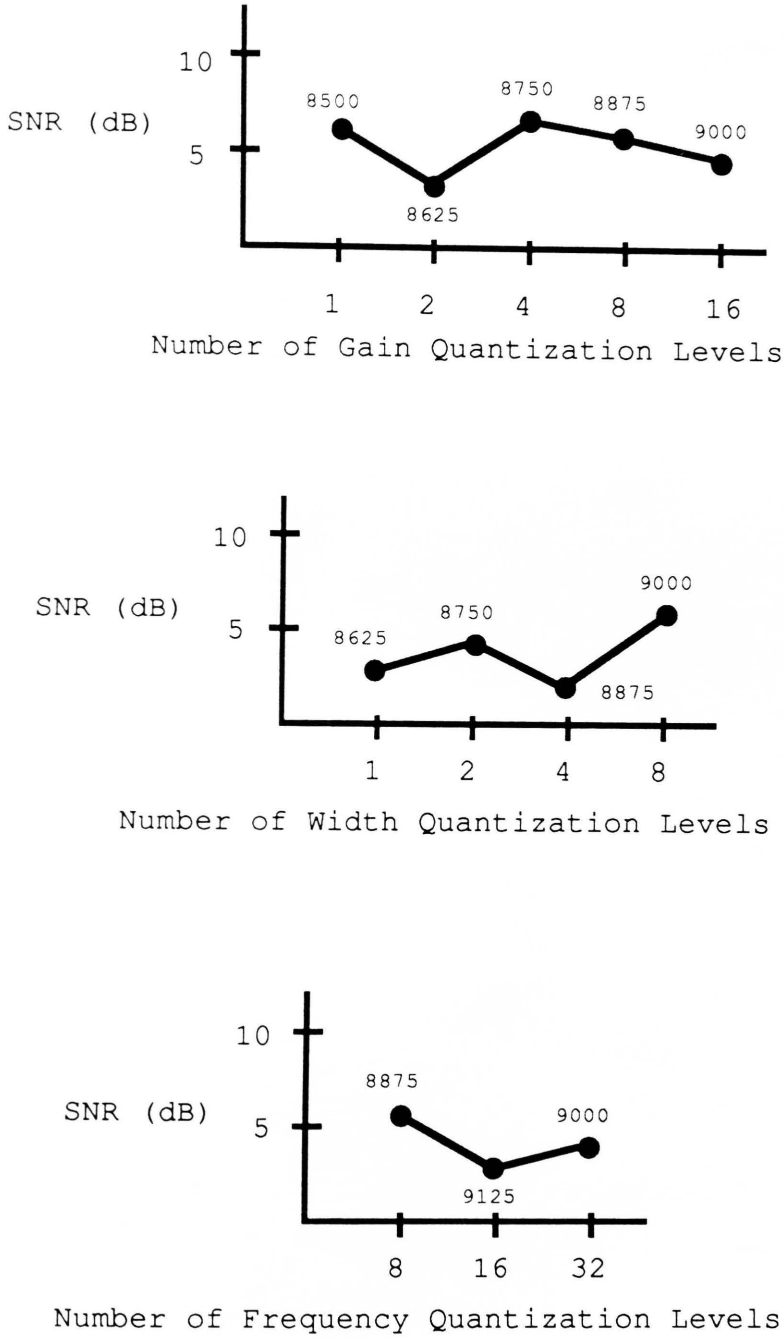



Figure 26. Effects of Gain, Width, and Start Frequency Ranges on the Signal to Noise Ratio. Male Speaker. Bit Rates Shown in Small Type.

degradation of the waveform. A similar result is obtained for a male speaker, as shown in Figure 26.

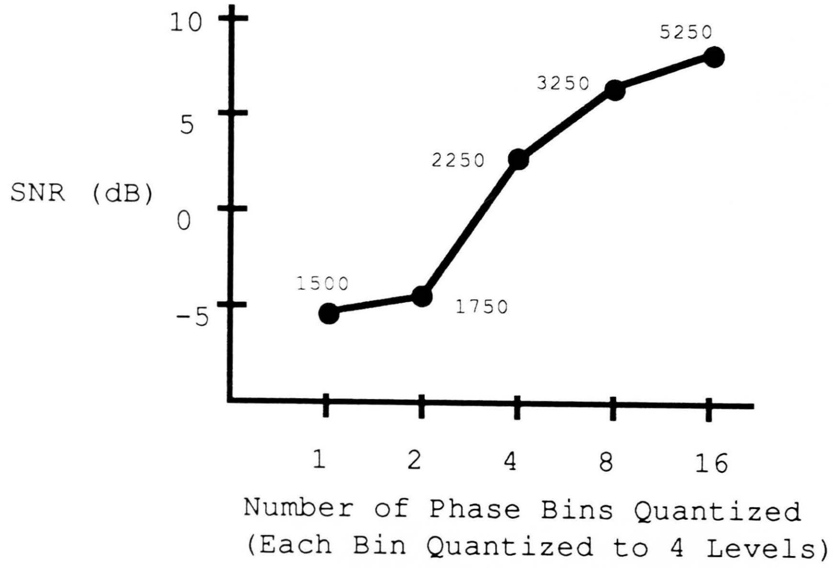
## Number of Phase Bins and Phase Quantization

The number of phase bins that are quantized, and not given random values, radically affects the SNR. Figure 27 shows the SNR of the reconstructed female speech for various numbers of phase bins quantized to four or eight levels. As the number of quantized phase bins increases, the SNR increases. The gain in SNR is even more dramatic when the phases are quantized to sixteen values, as shown in Figure 28 for the female speaker and in Figure 29 for the male speaker.

A summary plot of the effects of various numbers of quantized phase bins for the female speaker is given in Figure 30. In this plot, it is seen that the quantization of phase bins is not very important when only one or two phase bins are quantized. However, as more phase bins are quantized, the formant frequencies are also quantized, causing a dramatic jump in the SNR when four or more phase bins are quantized. Also, as more phase bins are quantized, the number of quantization levels becomes more influential on the SNR.

## Number of Correctors per Frame

The number of correctors that are generated per frame determines, to a large degree, the SNR. Consider



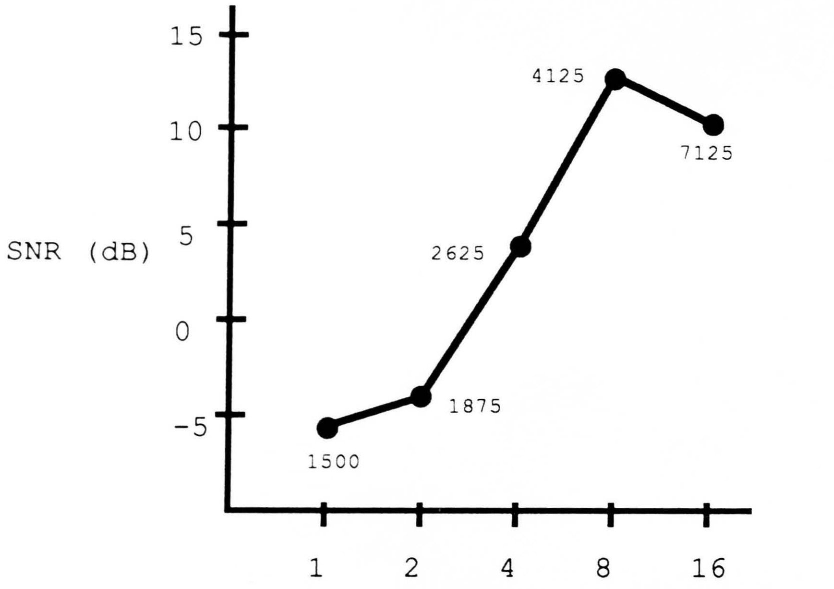




Figure 27. Effects of Number of Phases Quantized to Four or Eight Levels. Female Speaker. Bit Rates Shown in Small Type.

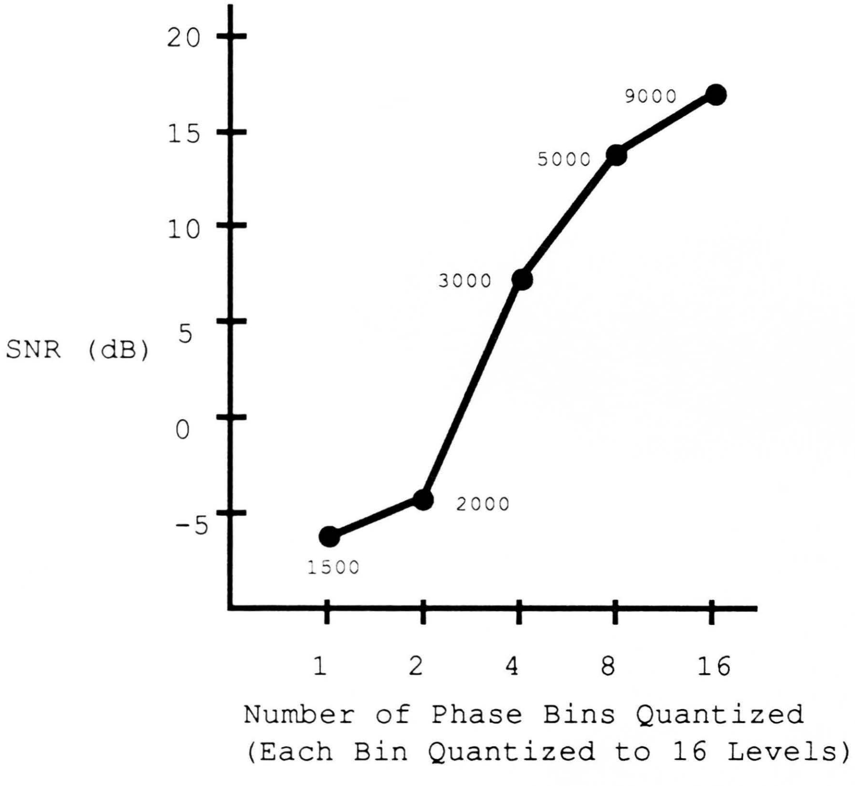



Figure 28. Effects of Number of Phases Quantized to Sixteen Levels. Female Speaker. Bit Rates in Small Type.

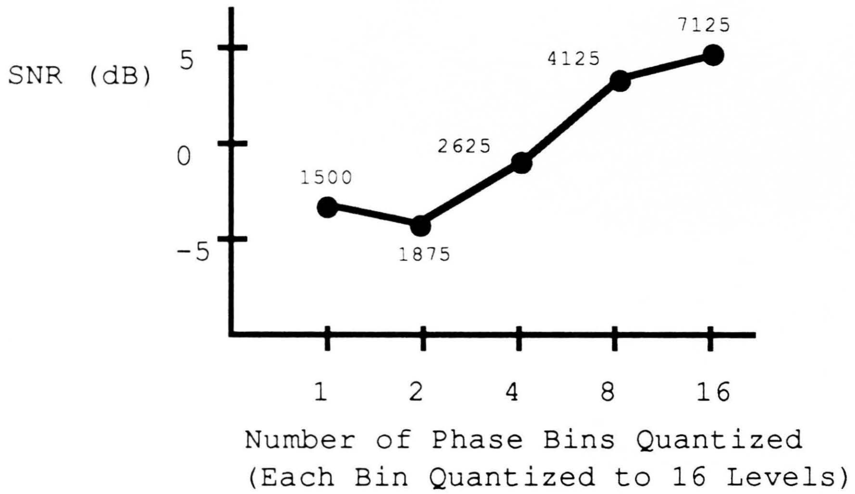



Figure 29. Effects of Number of Phases Quantized to Sixteen Levels. Male Speaker. Bit Rates Shown in Small Type.

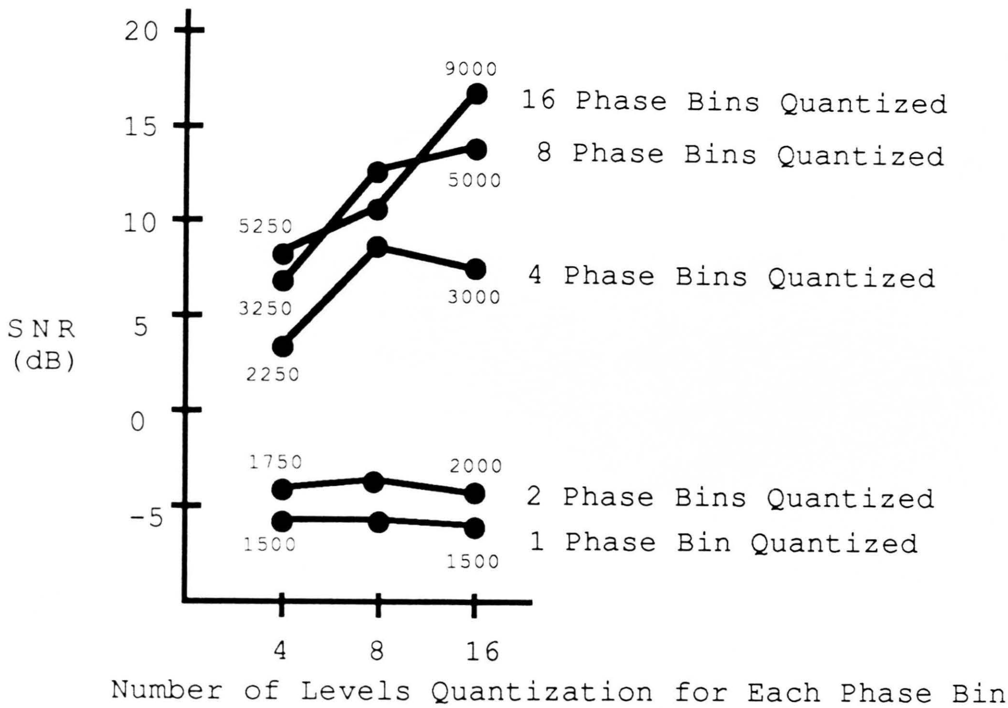



Figure 30. Summary of Signal to Noise Ratio Versus the Number of Phase Bins Quantized and the Number of Phase Quantization Levels. Female Speaker. Bit Rates Shown in Small Type.

the results shown in Figure 31. As the number of correctors in each frame increases, the SNR increases asymptotically to a value of around 20 dB. For this speech segment, very little is gained by increasing the number of correctors per frame greater than three; the additional bits could be better used for other parameters. Figure 32 displays PSC waveforms with one, two, and five correctors per frame; the original waveform is shown in Figure 23.

An interesting characteristic of PSC is shown in Figure 33, where the gain parameter probability density functions are plotted for different numbers of correctors per frame. As the number of correctors per frame increases, the gain values that are picked begin to cluster around the lowest values. This is understandable because after the first corrector in each frame has been computed, each additional corrector in the frame is used to refine the spectrum to a lesser extent. Therefore, smaller gains are chosen.

## Frame Length

It is difficult to obtain a relationship between frame lengths and performance because bit rates change for different lengths, numbers of correctors, and numbers of phases quantized. However, the performance appears to be constant with frame lengths of 64, 128, 256, and 512

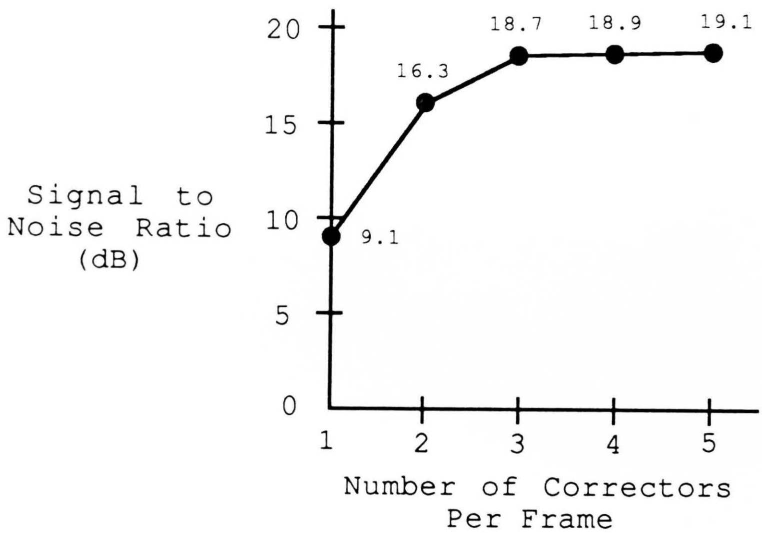



Figure 31. Effect of Number of Correctors Per Frame on the Signal to Noise Ratio. Female Speaker. Measured Signal to Noise Ratio Values Shown in Small Type.

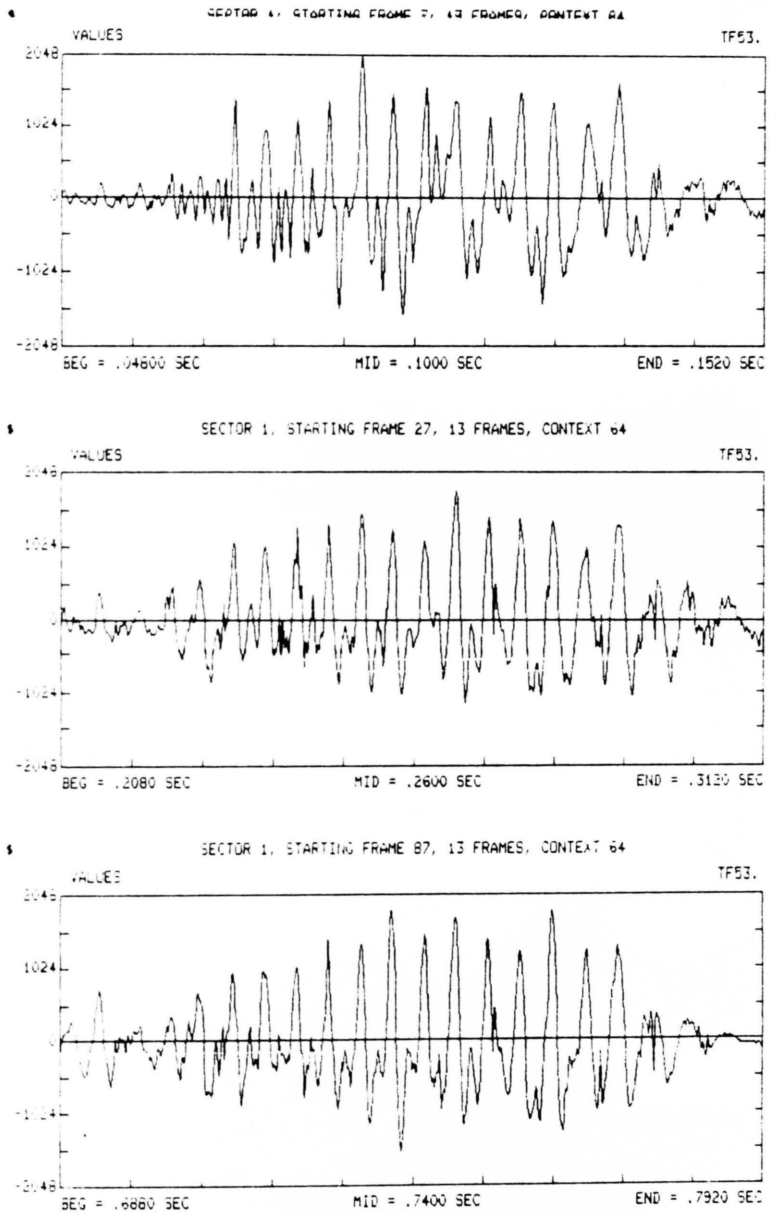



Figure 32. PSC Waveforms of Figure 23 With One, Two, and Five Correctors Per Frame.

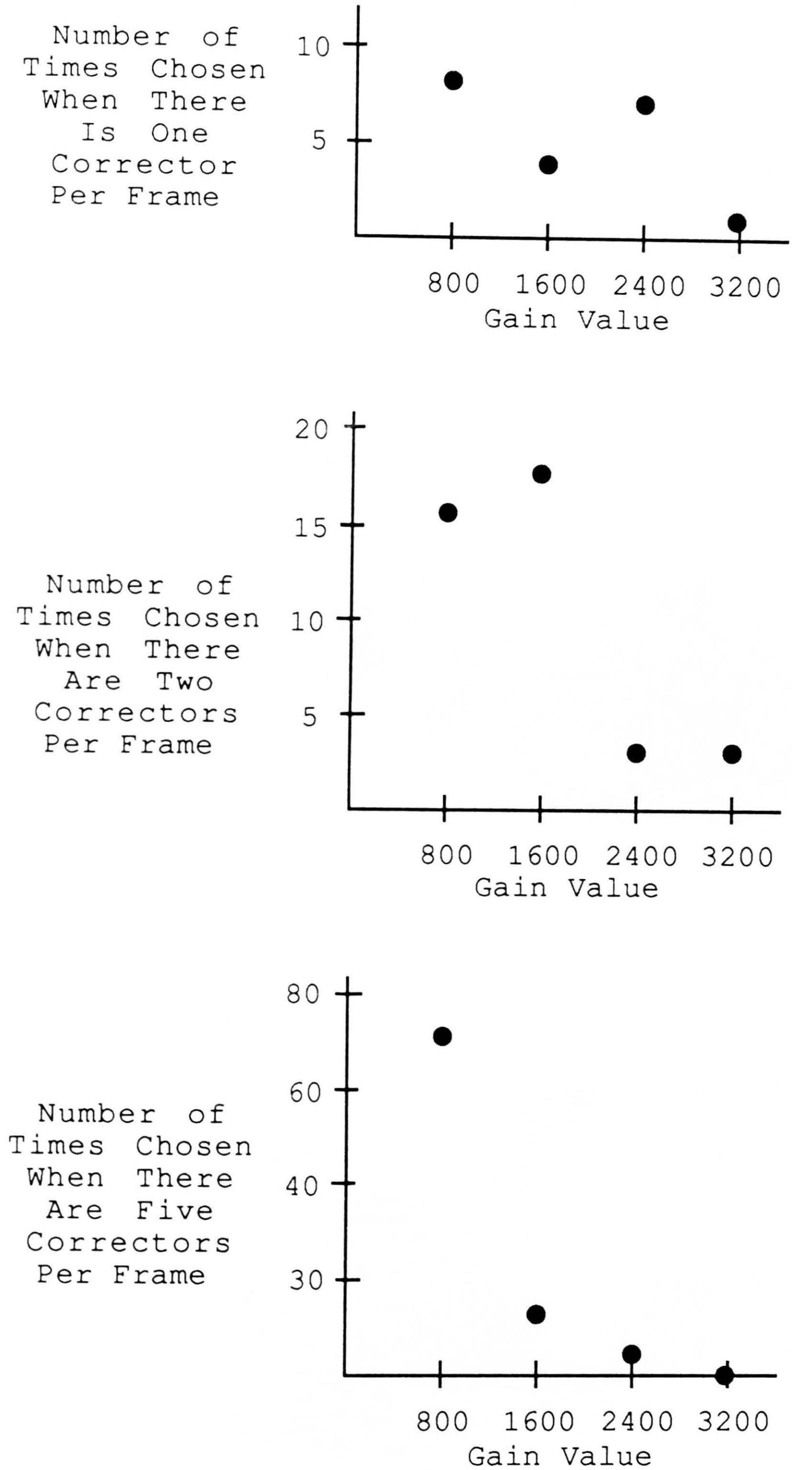



Figure 33. Gain Probability Density Functions for Different Numbers of Correctors Per Frame.

samples per frame, given approximately equivalent bit rates. Therefore, frame length is not an especially sensitive parameter in PSC.

## Subjective Measurements

Informal listening tests indicate that SNR measurements are an accurate means to measure the subjective quality of PSC speech. As the SNR increases, the quality of the speech rises accordingly. At 1200 bits per second, the speech is severely degraded but is mostly intelligible. Synthetic quality speech is obtained at between 2400 and 4800 bits per second. Speech that approaches communication quality is obtained at 9600 bits per second.

## Performance in the Presence of Noise

The performance of PSC in the presence of noise is plotted in Figure 34. Random bit errors, at controlled bit error percentage rates, were forced into the simulated corrector word before the speech was reconstructed. The drop in SNR for increasing bit error rates appears to be linear on the logarithmic SNR plot. The slope is around -1.7 dB per percent bit error rate. Displays of actual PSC waveforms with no error, two percent bit errors, and eight percent bit errors are shown in Figure 35. The original speech waveform is shown in Figure 23. As is seen, the waveform is still recognizable with a bit error

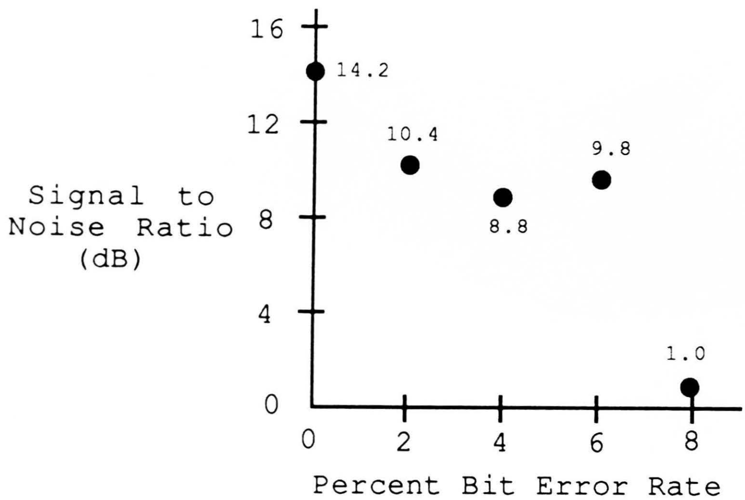



Figure 34. Effect of Bit Error Rate on the Signal to Noise Ratio. Female Speaker. Measured Signal to Noise Values in Small Type.

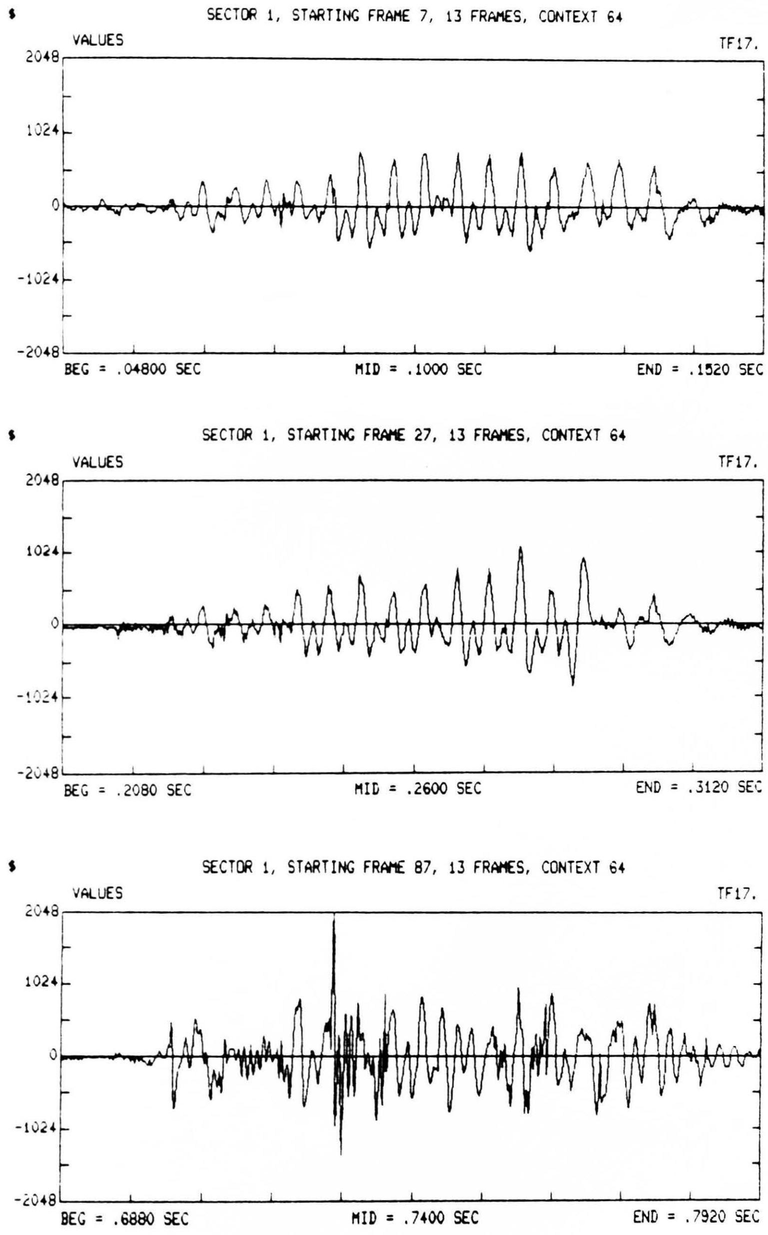



Figure 35. PSC Waveforms of Figure 23 With One, Two, and Eight Percent Bit Error Rate; 8750 Bits Per Second.

rate of two percent. However, a bit error rate of eight percent causes the waveform to lose most of its original character.

## Comparison of PSC to LPC-10

In its current state, PSC does not match the performance of LPC-10 with respect to the quality of the speech coded at 2400 bits per second. LPC-10 provides speech that is very intelligible and fairly natural sounding. At 2400 bits per second, PSC provides intelligible speech; but the signal contains popping noises and other discontinuities that are not present in reconstructed LPC-10 speech. There is potential, however, for significant improvements in the performance of PSC. Only a small subset of all possible avenues of research in PSC have been explored.

## Directions for Further Research

The results of the research performed to date shed some light on which directions for future experimentation might be intelligent choices. Goals of future research could be a reduction in the required bit rate for a given speech quality, better speech quality for a given bit rate, or both.

One change to the algorithm that might be fruitful is to always quantize the phase bins that correspond to the frequency components of the sampled speech with the highest energy levels. Although this is obviously desirable, it has the disadvantage that the frequency values which contain the highest energies must be transmitted along with the phase information. An easy method to resolve the problem is to quantize the phase bins that correspond to the highest evergy levels of the reference spectrum; the receiver can calculate for itself the frequency bins with the highest energy levels.

Some method for grading PSC systems needs to be developed. There is such a wide range of choices for parameters and bit rates that it is difficult to measure the quality of two different PSC systems if they require different bit rates. Perhaps a "SNR ÷ transmission rate" ratio would be a fair measurement.

One of the limitations of the current PSC design is that it does not effectively exploit the correlation of the phases from frame to frame. It is easier to take advantage of the magnitude correlation because high correlation means that the magnitude does not change appreciably between frames. Even with high correlation, however, phases have very different values each frame. Another problem with PSC is that it causes phase discontinuities at each frame boundary due to the phase quantization process. This results in noise which is objectionable in speech.

There are two possibilities for overcoming these problems. First, predictors can be computed which predict the value of the phase of a frequency component in future frames based on past frame values. This might require less bits than the current method of transmitting a quantized version of the phase each frame. Second, a hybrid coder system can be tried in which PSC is used to encode the spectrum magnitude and a suitable vocoder system is used to reconstruct the phases.

As an example of a hybrid system, PSC can be used in conjunction with LPC. The hybrid system can take advantage of the strengths of each type of coder. PSC can be used to encode the spectrum envelope. Pitch detectors and voiced/non-voiced decisions can be made with a subset of an LPC algorithm. The potential advantage here is that PSC provides a better estimate of the spectrum envelope at a lower bandwidth cost, and LPC (and other vocoders) reconstruct the phase with no discontinuities while requiring no phase information to be sent over the channel.

Experimentation with different frame allocation schemes should result in some performance gains. Some parameter values require updates more often than others. Center frequency and width, for example, probably require updates each frame. Gain, on the other hand, might not require updates at the frame rate.

# APPENDICES

# APPENDIX 1

## PSC SIMULATION SOFTWARE DESCRIPTION

Several software programs are used to simulate PSC. These programs fall into one of three categories: ILS software; PSC algorithm and related software; and data format translation programs. Figure 36 gives an overview of the procedure to be followed for typical PSC simulation exercises. The first step consists of obtaining a speech sample for processing; ILS software is used for this step. A polar FFT is performed on the speech segment, and ILS provides the software for this. The spectral file must be converted to an ASCII format before PSC processing can be performed. To do this, REC_FFT_TO_DAT.FOR is run.

PSC.FOR is the program to perform the actual PSC algorithm on the ASCII version of a polar FFT of the speech signal. PSC.FOR prompts the user for parameter values. PSC.FOR requires as little as a few seconds to run if the parameters are simple and the speech sample is short. If the parameters are complex and there are many samples (on the order of 500), then several hours might be required. The PSC file that is generated by PSC.FOR must be reconverted back into ILS format with the FFT_TO_DAT_SAM.COM command file. An inverse FFT is computed with ILS, and the reconstructed speech is ready for display or listening. Additionally, the SNR of the time domain speech waveform can be computed. To compute the SNR, the time domain waveform must first be translated into ASCII format. REC_SAM_TO_DAT.FOR performs the

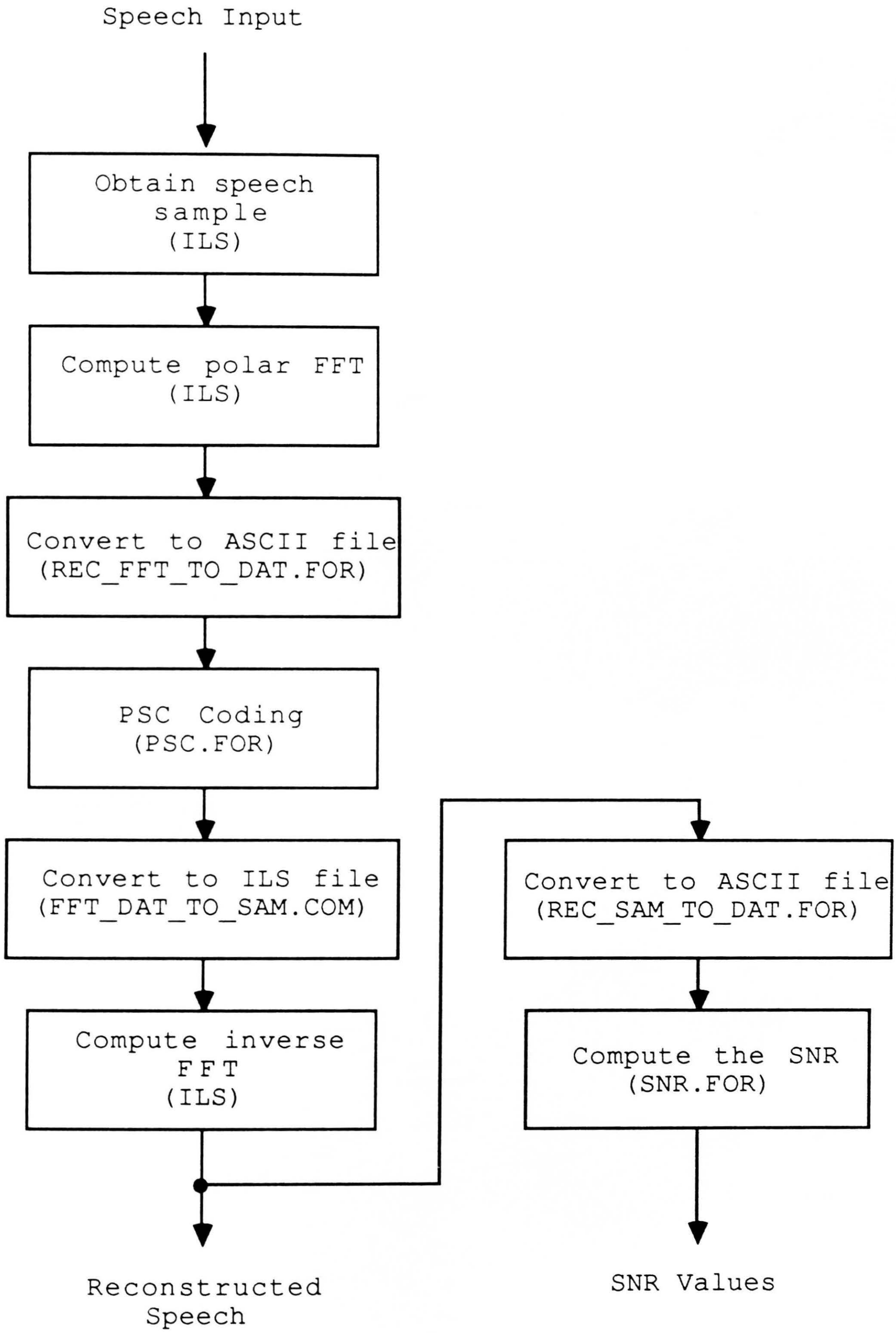



Figure 36. Procedure for PSC Simulation, Showing the Associated Software Tools Required.

translation. The SNR is calculated with the SNR.FOR program.

# APPENDIX 2

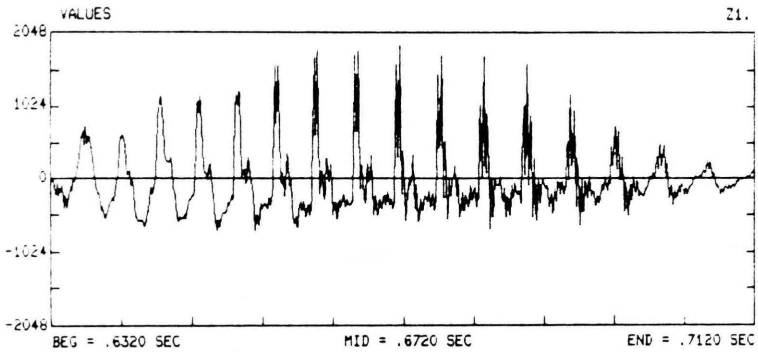
# PSC SIMULATION SAMPLE RUNS

Two sample runs of PSC simulation are presented here to aid the user in the use of the simulation tools. In the first sample run, the input speech segment contains 100 frames with 64 samples per frame. The sampling rate is 8000 Hz. A display of the speech segment is shown. When PSC.FOR is run, two correctors per frame are chosen. Each corrector quantizes gain (magnitude) to four levels, width to four levels, and phase values to eight levels. Sixteen start frequencies are available, and the first five frequency bins have their phases quantized. The resulting transmission rate is 3750 bits per second. The reconstructed speech segment is shown.

The speech segment used in the second run contains 50 frames with 64 samples per frame. All 50 frames are shown displayed. The sampling rate is 3000 Hz. In this run, only one correcter per frame is allowed. Each corrector quantizes gain to two levels, width to four levels, and phases to four levels. Eight frequencies are available for start frequencies, and the lowest four phase bins are quantized. The transmission rate is 644 bits per second. A display of the first 30 frames of the reconstructed speech is shown.

```
$ fil 1
Z1.                                          SAMPLED DATA
    20. DK BLKS.      100. FRAMES,         64 PT/FR
SAMPLE RATE =   8000 HZ
PRIMARY FILE
$ dsp e80,10
```

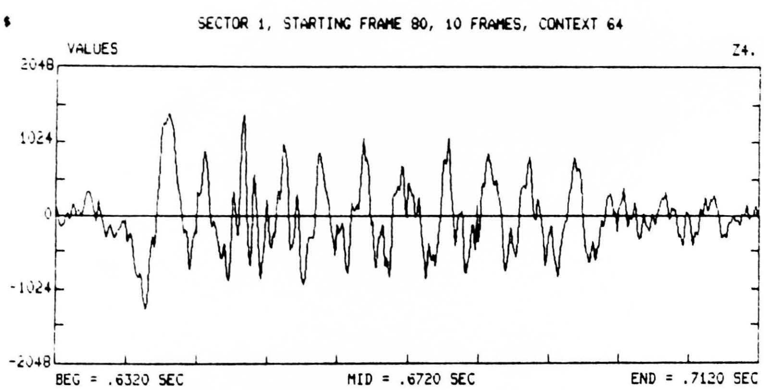
$ SECTOR 1, STARTING FRAME 80, 10 FRAMES, CONTEXT 64



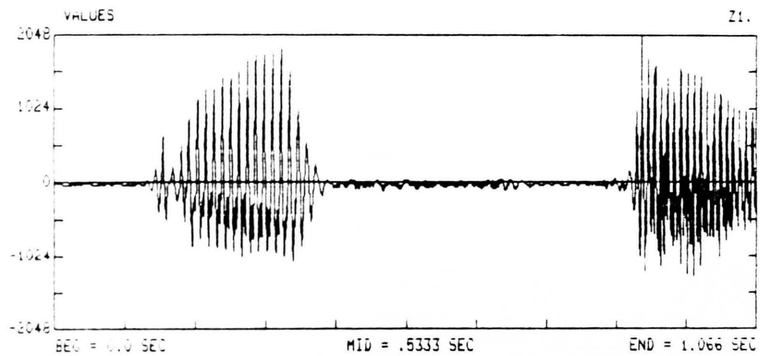

```
$ fil s2/opn s/sre 1,-1/fil 2/fil s3/opn s/fft pl,,6
Z2.                                          DOES NOT EXIST
SECONDARY FILE
$ OPN S
$ SRE 1,-1
Z2.                                          RECORD      8 STORED
Z2.                                          RECORD     16 STORED
Z2.                                          RECORD     24 STORED
Z2.                                          RECORD     32 STORED
Z2.                                          RECORD     40 STORED
Z2.                                          RECORD     48 STORED
Z2.                                          RECORD     56 STORED
Z2.                                          RECORD     64 STORED
Z2.                                          RECORD     72 STORED
Z2.                                          RECORD     80 STORED
Z2.                                          RECORD     88 STORED
Z2.                                          RECORD     96 STORED
Z2.                                          RECORD    100 STORED
$ FIL 2
Z2.                                          RECORD DATA
      3 DK BLKS.   100 RECORDS
PRIMARY FILE
$ FIL S3
Z3.                                          DOES NOT EXIST
SECONDARY FILE
$ OPN S
$ FFT PL,,6
Z3.                                          RECORD     32 STORED
Z3.                                          RECORD     64 STORED
Z3.                                          RECORD     96 STORED
Z3.                                          RECORD    100 STORED
$ run rec fft to dat
Enter the original record file to be processed z3.
Enter the ASCII file name                          z.dat
Enter the number of record frames                  100
Enter the frame length; CTX=                       64
FORTRAN STOP
$ run psc
Polar FFT data file to be processed:                        z.dat
P.S.C. file name:                                           zpsc.dat
Samples per second:                                         8000
Number of samples per frame:                                64
Number of frames:                                           100
Number of correctors per frame (1-10):                      2
# Levels magnitude is quantized to (2**N,N=0-6):            4
# Levels corrector width is quantized to (1,2,4,8):         4
# Frequencies searched for corr freq (2**N,N=0-8):          16
# Levels phase is quantized to (2**N,N=0-6):                8
Highest phase bin to be quantized (N=1-129):                5
Number of PSC bits per second is      3750.000
FORTRAN STOP
```

```
$ @ fft dat to sam
WHAT IS THE NAME OF THE INPUT DATA SPECTRAL FILE?: zpsc.dat
WHAT IS THE NAME OF THE OUTPUT SAMPLED FILE?: z4.
Alphabetic characters set to:  EX
$ FIL DEY100,,3
EX100.                                  DELETED
EX101.                                  DELETED
EX102.                                  DELETED

EX100.                                  DOES NOT EXIST
PRIMARY FILE
$ OPN
DELETE-I-FILDEL, USER01:[DCHAPMAN.EEL6502]WRTIN.DAT;1 deleted (21 blocks)
EX100.                                  RECORD      1 STORED
EX100.                                  RECORD      2 STORED
EX100.                                  RECORD      3 STORED
EX100.                                  RECORD      4 STORED
EX100.                                  RECORD      5 STORED
EX100.                                  RECORD      6 STORED
EX100.                                  RECORD      7 STORED
EX100.                                  RECORD     96 STORED
EX100.                                  RECORD     97 STORED
EX100.                                  RECORD     98 STORED
EX100.                                  RECORD     99 STORED
EX100.                                  RECORD    100 STORED
$ FIL S101
EX101.                                  DOES NOT EXIST
SECONDARY FILE
$ OPN S1
$ FFT 1
EX101.                                  RECORD     32 STORED
EX101.                                  RECORD     64 STORED
EX101.                                  RECORD     96 STORED
EX101.                                  RECORD    100 STORED
$ FIL 101
EX101.                                  RECORD DATA
    73 DK BLKS,  100 RECORDS
PRIMARY FILE
$ FIL S102
EX102.                                  DOES NOT EXIST
SECONDARY FILE
$ SSD
SCALE FACTOR= 0.1028367E+01,  MAX ABS VALUE= 0.1990535E+04
   100 FRAMES STORED
$ fil sam/fil 4
Alphabetic characters set to:  Z
$ FIL 4
Z4.                                     SAMPLED DATA
    [illegible] DK BLKS,    100. FRAMES,    64 PT/FR
SAMPLE RATE = 8000 HZ
PRIMARY FILE
$ [illegible] 80,10
```



```
       SAMPLED DATA
   14 DR BLKS,      50. FRAMES,     64 PT/FR
SAMPLE RATE -  3000 HZ
PRIMARY FILE
```




```
          opn s/ste 1,-1/fil 2/fil s3/opn s/fft pl,,6
                                          DOES NOT EXIST
SECONDARY FILE
  OPN S
  ORE 1,-1
                                          RECORD      8 STORED
                                          RECORD     16 STORED
                                          RECORD     24 STORED
                                          RECORD     32 STORED
                                          RECORD     40 STORED
                                          RECORD     48 STORED
                                          RECORD     50 STORED

                                          RECORD DATA
              50 RECORDS
PRIMARY FILE

                                          DOES NOT EXIST
SECONDARY FILE

  FFT PL,,6
                                          RECORD     32 STORED
                                          RECORD     50 STORED
     rec fft to dat
Enter the original record file to be processed z3.
Enter the ASCII file name                          z.dat
Enter the number of record frames                  50
Enter the frame length: CTX=                       64
FORTRAN STOP

Enter FFT data file to be processed:                    z.dat
     file name:                                         zbsc.dat
Samples per second:                                     3000
Number of samples per frame:                            64
Number of frames:                                       50
Number of correctors per frame (1 10):                  1
            magnitude is quantized to (2**N,N=0-6):    2
          corrector width is quantized to (1,2,4,8):   4
          frequencies searched for corr freq (2**N,N=0-8): 8
          phase is quantized to (2**N,N=0-6):          4
          phase bin to be quantized (N=1 12):          4
Number of FSC bits per second is      644.0000
FORTRAN STOP
```

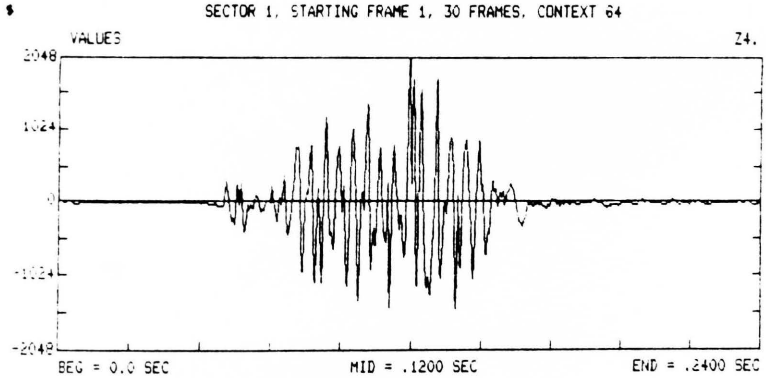
```
$ @ fit dat to sam
WHAT IS THE NAME OF THE INPUT DATA SPECTRAL FILE?: zpsc.dat
WHAT IS THE NAME OF THE OUTPUT SAMPLED FILE?: z4.
alphabetic characters set to:  Ex
$ FIL DE:100,,3
EX100.                              DELETED
EX101.                              DELETED
EX102.                              DELETED

EX103.                              DOES NOT EXIST
PRIMARY FILE
$ GEN
%DELETE-I-FILDEL, USER0:[DCHAPMAN.EELG50]IWRTIN.DAT;1 deleted (3 blocks)
EX100.                              RECORD      1 STORED
EX100.                              RECORD      2 STORED
EX100.                              RECORD      3 STORED
EX100.                              RECORD      4 STORED
EX100.                              RECORD     45 STORED
EX100.                              RECORD     46 STORED
EX100.                              RECORD     47 STORED
EX100.                              RECORD     48 STORED
EX100.                              RECORD     49 STORED
EX100.                              RECORD     50 STORED
$ FIL EX101
EX101.                              DOES NOT EXIST
SECONDARY FILE



EX101.                              RECORD     32 STORED
EX101.                              RECORD     50 STORED
$ FIL 101
EX101.                              RECORD DATA
    47 DK BLKS.     50 RECORDS
PRIMARY FILE
$ FIL S102
EX102.                              DOES NOT EXIST
SECONDARY FILE
$ SAM
SCALE FACTOR= 0.8592300E+00,  MAX ABS VALUE= 0.2382365E+04
   50 FRAMES STORED
$ fit sam/fil 4
alphabetic characters set to:  Z
$ FIL 4
Z4.                                 SAMPLED DATA
   13 DK BLKS.     50. FRAMES,     64 PT/FR
SAMPLE RATE =  8000 HZ
PRIMARY FILE
$ disp el, 1
$
$
$ disp el,30
$
```

SECTOR 1, STARTING FRAME 1, 30 FRAMES, CONTEXT 64



# APPENDIX 3

# PSC PERFORMANCE EXAMPLES

The first comparison was made between PSC and PCM for roughly equivalent intelligibility qualities. The PCM reference algorithm used eight-bit samples at a rate of 10,000 samples per second. The two algorithms provided equivalent intelligibility. The PSC speech produced more audible noise in the form of "clicks" and "pops." The PCM speech had no audible distortion. Table 6 gives the parameters of the two algorithms used in the first comparison.

TABLE 6

PARAMETERS USED IN ALGORITHM COMPARISON 1

| PARAMETER | POLAR SPECTRUM CODING | PULSE CODE MODULATION |
|---|---|---|
| Sample Rate | 10,000/sec | 10,000/sec |
| Filter Cutoff | 4.91 KHz | 4.91 KHz |
| Bit Rate | 28,438 bps | 80,000 bps |
| Samples Per Frame | 64 | - |
| Correctors Per Frame | 8 | - |
| Levels Corr. Magnitude | 4 | - |
| Levels Corr. Width | 8 | - |
| Bins Covered by Corr. | 32 | - |
| Phase Quant. Levels | 8 | - |
| Phase Bins Quantized | 32 | - |

A second comparison was made between PSC and PCM for roughly equivalent intelligibility qualities. The PCM reference algorithm used eight-bit samples at a rate of 2500 samples per second. The two algorithms provided equivalent intelligibility. The PSC speech produced more audible noise in the form of "clicks" and "pops." Both algorithms produced distortion due to the low sampling rate and filter cutoff frequency. Table 7 gives the parameters of the two algorithms used in the second comparison.

TABLE 7

PARAMETERS USED IN ALGORITHM COMPARISON 2

| PARAMETER | POLAR SPECTRUM CODING | PULSE CODE MODULATION |
|---|---|---|
| Sample Rate | 2500/sec | 2500/sec |
| Filter Cutoff | 1.24 KHz | 1.24 KHz |
| Bit Rate | 10,530 bps | 20,000 bps |
| Samples Per Frame | 64 | - |
| Correctors Per Frame | 16 | - |
| Levels Corr. Magnitude | 4 | - |
| Levels Corr. Width | 8 | - |
| Bins Covered by Corr. | 32 | - |
| Phase Quant. Levels | 8 | - |
| Phase Bins Quantized | 32 | - |

A third comparison was made between PSC and PCM for roughly equivalent transmission bit rates. The PCM reference algorithm used eight-bit samples at a rate of 2500 samples per second. The PSC speech was more intelligible than the PCM speech, but produced audible noise in the form of "clicks" and "pops." The PCM speech had distortion due to the low sample rate and filter cutoff frequency, and was less intelligible. Table 8 gives the parameters of the two algorithms used in the third comparison.

TABLE 8

PARAMETERS USED IN ALGORITHM COMPARISON 3

| PARAMETER | POLAR SPECTRUM CODING | PULSE CODE MODULATION |
|---|---|---|
| Sample Rate | 10,000/sec | 2500/sec |
| Filter Cutoff | 4.91 KHz | 1.24 KHz |
| Bit Rate | 20,280 bps | 20,000 bps |
| Samples Per Frame | 64 | - |
| Correctors Per Frame | 6 | - |
| Levels Corr. Magnitude | 4 | - |
| Levels Corr. Width | 8 | - |
| Bins Covered by Corr. | 32 | - |
| Phase Quant. Levels | 8 | - |
| Phase Bins Quantized | 22 | - |

# APPENDIX 4

# PSC SIMULATION PROGRAM LISTING

```
C       PSC.  This program takes a data file and performs Polar
C       Spectrum Coding on it.

        dimension TRY(514), S(50000), REF(257), ERR(257), PHASE(257)
        dimension ISG(256), ISQ(256), ISF(256), IRF(256)
        dimension TEMP(257), COR(257), PBIN(64)
        dimension IG(64), Q(8), IF(256)
        character*16 NAMEI,NAMEO
        integer*4 IFRAME, RUNNUM, SEED, IGQUANT, IQQUANT, IFQUANT, IPQUANT
        integer*4 IMAXBIN, KEEPBIN1, NSAMPLES
        integer*4 MAGPHASE, NMAGS, NPHASES, NCORRECTOR, NSAMPLERATE

110     format(A16)
111     format(A,$)
140     format(I6)
        write(6,111)' Polar FFT data file to be processed:              '
        read(5,110)NAMEI
        write(6,111)' P.S.C. file name:                                 '
        read(5,110)NAMEO
        write(6,111)' Samples per second:                               '
        read(5,140)NSAMPLERATE
        write(6,111)' Number of samples per frame:                      '
        read(5,140)NSAMPLES
        write(6,111)' Number of frames:                                 '
        read(5,140)IFRAME
        write(6,111)' Number of correctors per frame (1-10):            '
        read(5,140)NCORRECTOR
        write(6,111)' # Levels magnitude is quantized to (2**N,N=0-6):  '
        read(5,140)IGQUANT
        write(6,111)' # Levels corrector width is quantized to (1,2,4,8): '
        read(5,140)IQQUANT
        write(6,111)' # Frequencies searched for corr freq (2**N,N=0-8): '
        read(5,140)IFQUANT
        write(6,111)' # Levels phase is quantized to (2**N,N=0-6):      '
        read(5,140)IPQUANT
        write(6,111)' Highest phase bin to be quantized (N=1-129):      '
        read(5,140)IMAXBIN

        open (unit=1,name=NAMEI,status='OLD',form='FORMATTED',readonly)
        open (unit=2,name=NAMEO,status='NEW',form='FORMATTED')
        XG=IGQUANT        !change to real value for LOG function
        XQ=IQQUANT
        XP=IPQUANT
        XF=IFQUANT
        KEEPBIN1=0
        if (IMAXBIN.GE.1) KEEPBIN1=1
        !bits per frame for mag, including sign of corrector(s)
        XBITS=(1+(LOG(XG)+LOG(XQ)+LOG(XF))/LOG(2.0))*NCORRECTOR
        !total bits per frame, including 1 bit phase bin no. 1
        XBITS=XBITS+(LOG(XP)/LOG(2.0))*(IMAXBIN-1)+KEEPBIN1
        XBITS=XBITS*(NSAMPLERATE/NSAMPLES)
        write(*,*)'Number of PSC bits per second is ',XBITS
300     ICOUNT=1
        MAGPHASE=NSAMPLES+2      !# of data elements (mag & phase) per frame
        NMAGS=MAGPHASE/2
        NPHASES=MAGPHASE/2
        do I1=1,IFRAME                    !read in all frames
          read(1,*,end=900) (TRY(I), I=1,MAGPHASE)
          do J=1,MAGPHASE
            S(ICOUNT)=TRY(J)     !write data into array SBIG
            ICOUNT=ICOUNT+1
```

```
      end do
    end do
900 continue
    !get gain coefficients
    IDIV=LOG(XG)/LOG(2.0) +1
    IGNOMINAL=2048
    do I=1,IDIV
      IGNOMINAL=IGNOMINAL/2
    end do
    do I=1,IGQUANT
      IG(I)=IGNOMINAL
    WRITE(*,*) ' I, IG(I) ',I,IG(I)
      IGNOMINAL=IGNOMINAL*2
    end do
    Q(1)=1
      Q(3)=2
    Q(2)=3
        Q(5)=4
      Q(4)=5
        Q(6)=6
        Q(7)=8
        Q(8)=12
    do I=1,IPQUANT  !get phase coefficients
      PBIN(I)=((6.283185*I)/IPQUANT)-3.141593    !PBIN is 2PI divided by 8
    end do

    ISAVE=1
    ICOUNT=1
    do K=1,IFRAME            !expands the magnitude scale somewhat
      do J=1,NMAGS
        if (S(ICOUNT) .LT. 0.0) S(ICOUNT)=0.00001
        S(ICOUNT)=S(ICOUNT)**2.0
        ICOUNT=ICOUNT+2
      end do
    end do
    do I=1,NMAGS
      REF(I)=0      !initialize reference spectrum
    end do

    SEED=400003
    ICOUNT=1

    !DO PSC ON EACH FRAME
    do J=1,IFRAME   !do PSC on the phases of each frame
      do K=2,NPHASES        !randomize phase bins
        PHASE(K)=(RAN(SEED)*3.14159265*2)-3.14159265
      end do
      ICOUNT=ISAVE+1      !update phase of 1st freq bin with pi or 0
      if (KEEPBIN1.EQ.1) PHASE(1)=S(ICOUNT)
      IUPDATE=3
      if(IMAXBIN.LT.2) goto 25
      do I=2,IMAXBIN
        ICOUNT=ISAVE+IUPDATE         !update PHASE bin I
        PHASE(I)=S(ICOUNT)
        XMINDIFF=4.0          !initialize to higher than pi
        do K=1,IPQUANT        !find closest fit to actual phase
          XDIFF=ABS(PHASE(I)-PBIN(K))
          if (XDIFF .LT. XMINDIFF) MINDIFF=K
          if (XDIFF .LT. XMINDIFF) XMINDIFF=XDIFF
        end do
        PHASE(I)=PBIN(MINDIFF)       !quantize phase bin I
        IUPDATE=IUPDATE+2   !index to next phase bin in file
      end do
25    continue
      ICOUNT=ISAVE+1          !set ICOUNT to 1st phase of frame
      do K=1,NPHASES          !write PHASE into phase part of S
```

```
        S(ICOUNT)=PHASE(K)
        ICOUNT=ICOUNT+2
      end do

     do i2=1,NCORRECTOR     !do NCORRECTOR magnitude correctors each frame
      !do PSC on the magnitude in each frame
      ICOUNT=ISAVE
      do I=1,NMAGS
        ERR(I)=S(ICOUNT)-REF(I)        !calculate error spectrum
        ICOUNT=ICOUNT+2
      end do
      AREAMIN=1000000000
      NPZ=-1
      do IPZ=1,2                       !try poles, then zeros
        NPZ=NPZ*-1                     !toggles between poles, zeros
        do IQ=1,IQQUANT                !try each Q
          do IFREQ=1,IFQUANT           !try each frequency
            do IGAIN=1,IGQUANT         !try each gain
              do ITEMP=1,NMAGS         !initialize TEMP to ERR
                TEMP(ITEMP)=ERR(ITEMP)
              end do
              IENDCORR=IFREQ+Q(IQ)-1
              if(IENDCORR .GT. IFQUANT) IENDCORR=IFQUANT
              do M=IFREQ,IENDCORR    !add/sub corrector
                TEMP(M)=TEMP(M)-NPZ*IG(IGAIN)
              end do
              AREA=0
              do M=1,NMAGS       !get area under curve
                AREA=ABS(TEMP(M))+AREA
              end do
              if(AREA.LT.AREAMIN) then
                NEWPZ=NPZ
                NQVALUE=Q(IQ)
                NFREQ=IFREQ
                NG=IG(IGAIN)
                AREAMIN=AREA
              end if
            end do             !end gain loop
          end do               !end freq loop
        end do                 !end Q loop
      end do                   !end p/z loop

    WRITE(*,*) ' Frame,Corr,Q,F,G = ',J,i2,NQVALUE,NFREQ,NG

      IENDCORR=NFREQ+NQVALUE-1
      if(IENDCORR .GT. IFQUANT) IENDCORR=IFQUANT
      do I=NFREQ,IENDCORR              !add corrector to ref spectrum
        REF(I)=REF(I)+NEWPZ*NG
      end do
     end do

      ICOUNT=ISAVE
      do I=1,NMAGS  !copy REF into MAGPHASE-value S
        if (REF(I) .LT. 0.0) REF(I)=0.00001
        S(ICOUNT)=REF(I)**0.5  !compresses the scale back
        ICOUNT=ICOUNT+2
      end do
      ISAVE=ISAVE+MAGPHASE
21  end do
    ISTART=1
    IEND=MAGPHASE
    do K=1,IFRAME
      write(2,*) (S(I), I=ISTART,IEND)
      ISTART=ISTART+MAGPHASE
      IEND=IEND+MAGPHASE
    end do
    stop
    end
```

# APPENDIX 5

# SNR MEASUREMENT PROGRAM LISTING

```
C       Program SNR.FOR.
C       This program calculates the Signal to Noise Ratio
C       between a sampled data file created by the ILS
C       program and a Polar Spectrum Coding file.

        implicit none
        integer i,j,k,il,L,m,n,nctx,ixstart,icompare,iystart,irepeat
        integer iyend,nxstart,ncompare,nystart,mstart,mend
        character*16 NX,NY,NSNR
        real X(8000),Y(8000,50),R(8000,50),TRY(8000)
        real RVAR(50),XVAR,RMEAN(50),XMEAN,SNRdB(50),XENERGY(0:100)
        real YMEAN(50),YENERGY(100),SCALE,RVARTEMP

111     format(A,$)
110     format(A8)
140     format(I4)

        write(6,111)' Enter the X signal file:                              '
        read(5,110)NX
        write(6,111)' Enter the Y signal file:                              '
        read(5,110)NY
        write(6,111)' Enter the SNR output file:                            '
        read(5,110)NSNR
        write(6,111)' Starting frame of X file to be compared:              '
        read(5,140)NXSTART
        write(6,111)' Number of frames of X file to be compared:            '
        read(5,140)NCOMPARE
        write(6,111)' Number of frames in X file repeated in Y file: '
        read(5,140)IREPEAT
        write(6,111)' Starting frame of Y file to be compared:              '
        read(5,140)NYSTART
        write(6,111)' CTX =                                                 '
        read(5,140)NCTX

        open (unit=1,name=NX,status='OLD',form='FORMATTED',readonly)
        open (unit=2,name=NY,status='OLD',form='FORMATTED',readonly)
        open (unit=3,name=NSNR,status='NEW',form='FORMATTED')

        IXSTART=(NXSTART-1)*NCTX
        ICOMPARE=NCOMPARE*NCTX
        IYSTART=(NYSTART-1)*NCTX+1
        IREPEAT=IREPEAT*NCTX

        !Put data up to 1 frame before starting sample of X file
        ! in bit bucket
        if (NXSTART.GT.2) then
          read(1,*,end=901) (TRY(I), I=1,IXSTART)
        end if
901     continue
        !Get X data that will be used for comparison to Y data,
        ! including 1 frame before starting frame and 1 frame after
        ! ending frame
        read(1,*,end=900) (X(I), I=1,ICOMPARE)
        read(1,*,end=900) (X(I), I=1,NCTX)
        read(1,*,end=900) (X(I), I=1,NCTX)
900     continue

C       GET THE MEAN AND VARIANCE OF X, AND THE ENERGY IN EACH FRAME OF X
        K=1
        XMEAN=0
        do I=0.NCOMPARE+1
```

```
      XENERGY(I)=0
      do J=1,NCTX
        XENERGY(I)=XENERGY(I)+ABS(X(K))/NCTX
        XMEAN=XMEAN+X(K)
        K=K+1
      end do
    end do
    XMEAN=XMEAN/ICOMPARE
    XVAR=0
    do I=1,ICOMPARE
      X(I)=X(I)-XMEAN        !normalize X by the mean
      XVAR=XVAR+(X(I)**2)    !get the variance
    end do
    write(*,*)'           J        RVAR          SNR(dB) '
    WRITE(*,*)' XVAR,X(I) = ',XVAR,X(I)
    XVAR=XVAR/ICOMPARE

    IYEND=IYSTART+ICOMPARE
    !Get Y data that will be used in comparison -- 2-dimensional
    do J=1,27         !27 is # times X data is repeated in Y file
      read(2,*,end=902) (TRY(I), I=1,IREPEAT)
      I=1
      YMEAN(J)=0
      do K=IYSTART,IYEND
        Y(I,J)=TRY(K)
        YMEAN(J)=YMEAN(J)+Y(I,J)
        I=I+1
      end do
      YMEAN(J)=YMEAN(J)/ICOMPARE
      do I=1,ICOMPARE
        Y(I,J)=Y(I,J)-YMEAN(J)        !normalize Y by the mean
      end do
C     NORMALIZE Y TO THE SAME ENERGY LEVELS THAT ARE IN X
      K=1
      do I=1,NCOMPARE
        YENERGY(I)=0
        do L=1,NCTX
          YENERGY(I)=YENERGY(I)+ABS(Y(K,J))/NCTX
          K=K+1
        end do
      end do
      K=1
      M=1
      mstart=1
      mend=NCTX
      do 1-1,NCOMPARE
        SCALE=XENERGY(I)/YENERGY(I)
        do L=1,NCTX
          Y(K,J)=Y(K,J)*SCALE
          K=K+1
        end do
        write(3,*) (Y(L,J), L=mstart,mend)
        mstart=mstart+NCTX
        mend=mend+NCTX
      end do
    end do
902 continue


C   CALCULATE ERROR (R), MEANS, VARIANCES, AND SNR
C   X and Y already have zero mean, so R=X-Y will have zero mean

C   For each set of frames, find minimum variance of R by sliding X
C    relative to Y and calculating variance until minimum is found
    do J=1,27
      RVAR(J)=1000000000    !Big number used for first comparison
```

```
  do K=1,NCTX*2 !Slide X over 2-frame window
    L=K
    do I=1,ICOMPARE
      R(I,J)=X(L)-Y(I,J)
      L=L+1
    end do
    RVARTEMP=0
    do I=1,ICOMPARE
      RVARTEMP=RVARTEMP+(R(I,J)**2)
    end do
    if (RVARTEMP.LT.RVAR(J)) RVAR(J)=RVARTEMP
  end do
end do
do J=1,27
  RVAR(J)=RVAR(J)/ICOMPARE
end do
write(*,*)'           J       RVAR          SNR(dB) '
write(3,*)'           J       RVAR          SNR(dB) '
do J=1,27
  SNRdB(J)=10*LOG((XVAR/RVAR(J))+0.00000001)
  write(*,*)J,RVAR(J),SNRdB(J)
  write(3,*)J,RVAR(J),SNRdB(J)
end do
write(*,*) ' XVAR = ',XVAR
write(3,*) ' XVAR = ',XVAR

close(1)
close(2)
close(3)
stop
end
```

# APPENDIX 6

# ILS/ASCII TRANSLATION PROGRAM LISTINGS

There are three translation programs used in PSC simulations. REC_FFT_TO_DAT.FOR translates an ILS record of a polar FFT into ASCII code. FFT_DAT_TO_SAM.COM is a command file which translates an ASCII polar FFT file into a sampled ILS file. It is shown for 64 samples per frame. For other values of CTX, the "33" in the line that reads

```
wrt sp33,2,3000,8000 . . .
```

should be changed to a value of $(CTX \div 2)+1$. The REC_SAM_TO_DAT.FOR program translates an ILS record time domain file into ASCII code.

```
C       PROGRAM REC_FFT_TO_DAT. This program takes a record data file
C       created by ILS and converts it to an ASCII data file.

        dimension       TRY(128), DIN(50000), DOUT(50000)
        character*16    NAMEI,NAMEO
        integer*4       IFRAME

110     format(A)
111     format(A,$)
112     format(I4)
        write(6,111)' Enter the original record file to be processed '
        read(5,110)NAMEI
        write(6,111)' Enter the ASCII file name                        '
        read(5,110)NAMEO
        write(6,111)' Enter the number of record frames                '
        read(5,112)IFRAME
        write(6,111)' Enter the frame length; CTX=                     '
        read(5,112)NCTX

        open (unit=1,name=NAMEI,status='OLD',form='UNFORMATTED',readonly)
        open (unit=2,name=NAMEO,status='NEW',form='FORMATTED')
        K=1
        NFRAME2=IFRAME*2
        do I1=1,NFRAME2                 !read in record file to DIN()
          read(1,end=910) (TRY(I), I=1,128)
          do J=1,128                    !read in each record
            DIN(K)=TRY(J)               !set DIN() to record value
            K=K+1
          end do
        end do
910     continue

        J=1
        NDIN=385                        !skip header in DIN()
        NDOUT=1
        ITOT=(IFRAME-7)*128
        MASDIR=IFRAME/32        !there is a pointer record every 32 data records
        NDATALENGTH=32*((NCTX+2)+12)
              !32 frames of NCTX+14 each between each pointer record
        do k=1,MASDIR
C       N/2+1 PAIRS DATA STORED MAG, PHAS, AFTER 12-WORD HEADER PER RECORD
          do K2=1,NDATALENGTH
            if (J.GT.12.AND.J.LE.(NCTX+14)) DOUT(NDOUT)=DIN(NDIN)
            if (J.GT.12.AND.J.LE.(NCTX+14)) NDOUT=NDOUT+1
            NDIN=NDIN+1
            J=J+1
            if (J .GT. NCTX+14) J=1              !MODULO NCTX+14
          end do
          J=1                   !start modulo J over again
          NDIN=NDIN+68          !skip directory record
        end do

        ITOTFRAME=NDOUT/(NCTX+2)
        ISTART=1
        IEND=NCTX+2
        do K=1,ITOTFRAME
          write(2,*) (DOUT(I), I=ISTART,IEND)
          ISTART=ISTART+NCTX+2
          IEND=IEND+NCTX+2
        end do
        ITOTFRAME=(NDIN/(NCTX+14))-6
```

```
      ISTART=385
      IEND=NCTX+14+ISTART
      do K=1,ITOTFRAME
        write(3,*) (DIN(I), I=ISTART,IEND)
        ISTART=ISTART+NCTX+14
        IEND=IEND+NCTX+14
      end do
900   continue
      close(1)
      close(2)
      stop
      end
```

```
$ INQUIRE NAMEIN 'WHAT IS THE NAME OF THE INPUT DATA SPECTRAL FILE?'
$ INQUIRE NAMEOUT 'WHAT IS THE NAME OF THE OUTPUT SAMPLED FILE?'
$ fil anex/fil dey100,,3/opn
$ DEL WRTIN.DAT;*
$ COPY 'NAMEIN WRTIN.DAT
$ wrt sp33,2,3000,8000/fil s101/opn s1/fft 1/fil 101/fil s102/ssd
$ COPY EX102. 'NAMEOUT
```

```
C       PROGRAM REC_SAM_TO_DAT. This program takes a record data file
C       created from an ILS sampled file and converts it to an ASCII data file.

        dimension         TRY(128), DIN(50000), DOUT(50000)
        character*16      NAMEI,NAMEO
        integer*4         IFRAME

110     format(A)
111     format(A,$)
112     format(I4)
        write(6,111)' Enter the original record file to be processed '
        read(5,110)NAMEI
        write(6,111)' Enter the ASCII file name                      '
        read(5,110)NAMEO
        write(6,111)' Enter the number of record frames              '
        read(5,112)IFRAME
        write(6,111)' Enter the frame length; CTX=                   '
        read(5,112)NCTX

        open (unit=1,name=NAMEI,status='OLD',form='UNFORMATTED',readonly)
        open (unit=2,name=NAMEO,status='NEW',form='FORMATTED')
        K=1
        NFRAME2=IFRAME*2
        do I1=1,NFRAME2                  !read in record file to DIN()
          read(1,end=910) (TRY(I), I=1,128)
          do J=1,128                     !read in each record
            DIN(K)=TRY(J)                !set DIN() to record value
            K=K+1
          end do
        end do
910     continue

        J=1
        NDIN=385                         !skip header in DIN()
        NDOUT=1
        ITOT=(IFRAME-7)*128
        MASDIR=IFRAME/8         !there is a pointer record every 8 data records
        NDATALENGTH=8*(NCTX+12)
              !8 frames of NCTX+14 each between each pointer record
        do k=1,MASDIR
C       N/2+1 PAIRS DATA STORED MAG, PHAS, AFTER 12-WORD HEADER PER RECORD
          do K2=1,NDATALENGTH
            if (J.GT.12) DOUT(NDOUT)=DIN(NDIN)
            if (J.GT.12) NDOUT=NDOUT+1
            NDIN=NDIN+1
            J=J+1
            if (J .GT. NCTX+12) J=1              !MODULO NCTX+12
          end do
          J=1                     !start modulo J over again
          NDIN=NDIN+20  !skip directory record
        end do

        ITOTFRAME=NDOUT/NCTX
        ISTART=1
        IEND=NCTX
        do K=1,ITOTFRAME
          write(2,*) (DOUT(I), I=ISTART,IEND)
          ISTART=ISTART+NCTX
          IEND=IEND+NCTX
        end do
c       ITOTFRAME=(NDIN/(NCTX+14))-6
```

```
c       ISTART=385
c       IEND=NCTX+14+ISTART
c       do K=1,ITOTFRAME
c         write(3,*) (DIN(I), I=ISTART,IEND)
c         ISTART=ISTART+NCTX+14
c         IEND=IEND+NCTX+14
c       end do
900     continue
        close(1)
        close(2)
        stop
        end
```

## LITERATURE CITED

Jayant, N. S., and Noll, Peter. 1984. Digital Coding of Waveforms: Principles and Applications to Speech and Video. Englewood Cliffs, N. J.: Prentice-Hall.

Kitawki, Nobuhiko; Itoh, Kenzo; Honda, Masaaki; and Kakehi, Kazuhiko. 1982. Comparison of Objective Speech Quality Measures for Voiceband Codecs. IEEE International Conference on Accoustics, Speech and Signal Processing (May): 1000-1003.

Oppenheim, Alan V., and Schafer, Ronald W. 1975. Digital Signal Processing. Englewood Cliffs, N. J.: Prentice-Hall.

Papamichalis, Panos E. 1987. Practical Approaches to Speech Coding. Englewood Cliffs, N. J.: Prentice-Hall.

Rabiner, Lawrence R., and Schafer, Ronald W. 1978. Digital Processing of Speech Signals. Englewood Cliffs, N. J.: Prentice-Hall.